

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: “Веб портал автомобільного салону”

Виконав: _____ Гончар Владислав Олександрович

Керівник: _____ доцент Моденов Юрій Борисович

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 студентом

Гончаром Владиславом Олександровичем

1. Тема роботи: " Веб портал автомобільного салону "

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: існуючі веб портали, що надають можливість перегляду та купівлі пропонованих продуктів.

4. Зміст пояснювальної записки: вступ, аналіз предметної області, опис та обґрунтування вибраного програмного забезпечення, етапи реалізації та тестування готового веб порталу.

5. Перелік обов'язкового ілюстративного матеріалу: рисунки, діаграми, а також слайди презентації доповіді у PowerPoint.

6. Календарний план-графік

| № з/п | Завдання | Термін виконання | Підпис керівника |
|-------|---|--------------------------|------------------|
| 1. | Формування теми дипломної роботи, постановка задачі та узгодження з дипломним керівником. | 12.10.2021 15.10.2021 | – |
| 2 | Формування структури розділів дипломної роботи | 16.10.2021 19.10.2021 | – |
| 3 | Формування та оформлення першої частини дипломної роботи | 20.10.2021 24.10.2021 | – |
| 4 | Збір науково-технічного матеріалу до другої частини дипломної роботи | 25.10.2021 31.10.2021 | – |
| 5 | Формування та оформлення другої частини дипломної роботи | 01.11.2021 07.11.2021 | – |
| 6 | Розробка програмної частини згідно з завдання дипломної роботи | 08.11.2021 17.11.2021 | – |
| 7 | Формування та оформлення третьої частини дипломної роботи | 18.11.2021 01.12.2021 | – |
| 8 | Формування звіту та графічних матеріалів | 02.12.2021 11.12.2021 | – |
| 9 | Підписання необхідних документів | 12.12.2021 14.12.2021 | – |
| 10. | Підготовка до захисту дипломної роботи | 15.12.2021 20.12.2021 | – |

7. Дата видачі завдання 10 жовтня 2021р.

Керівник дипломної роботи . _____ Юрій МОДЕНОВ.

Завдання прийняв до виконання _____ Владислав ГОНЧАР.

(підпис випускника)

(ПБ)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи "Веб портал автомобільного салону" містить: 87 сторінок, 18 рисунків, 4 таблиці, 33 використаних джерела.

За результатами дослідження сформульовані висновки щодо методології розробки сайту в процесі виконання роботи. Одержані результати можуть бути використані в подальшому перепроєктуванню сайту та обслуговуванні.

Мета дипломної роботи – розробка веб порталу автомобільного салону та подальше впровадження його в роботу.

Об'єкт дослідження – існуючі технології розробки та проектування веб застосунків на схожу тематику.

Предмет дослідження – веб-орієнтовані системи інтернет торгівлі.

Метод дослідження –моніторинг, аналіз, порівняння та розробка.

Результат проекту – веб портал автомобільного салону.

Ключові слова: JavaScript, PHP, CSS, ВЕБ-ПОРТАЛ, SQL, АРХІТЕКТУРА, ТЕСТУВАННЯ, АВТОМОБІЛЬНИЙ САЛОН.

ЗМІСТ

| | |
|---|----|
| ВСТУП..... | 6 |
| РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ | 10 |
| 1.1. Аналіз предметної галузі | 10 |
| 1.2 Аналіз платіжних систем | 22 |
| 1.3 Постановка задачі | 28 |
| ВИСНОВКИ ДО РОЗДІЛУ 2 | 30 |
| РОЗДІЛ 2. ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ..... | 31 |
| 2.1 Розмітка сторінок - HTML..... | 33 |
| 2.2 JavaScript | 36 |
| 2.3 Server pages..... | 39 |
| 2.4. Реалізація програмного забезпечення | 49 |
| ВИСНОВКИ ДО РОЗДІЛУ 2 | 60 |
| РОЗДІЛ 3. ДИЗАЙН ІНТЕРФЕЙСУ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ | 61 |
| 3.1. Структура веб порталу | 61 |
| 3.2. Програмна реалізація | 71 |
| 3.3 Оцінка розробленого програмного продукту | 75 |
| ВИСНОВКИ ДО РОЗДІЛУ 3 | 81 |
| ВИСНОВКИ..... | 82 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 85 |

ВСТУП

Сучасне суспільство неможливо уявити без інтернету, соціальних мереж та мобільних телефонів. У своєму творі хочеться поміркувати на цю тему, зважити всі переваги та недоліки інтернету і, звичайно, обміркувати його роль у житті кожного з нас.

На перший погляд, може здатися, що поява інтернету принесла одні плюси. І справді, завдяки інтернету людям стало простіше шукати потрібну інформацію. Якщо раніше школярам і студентам доводилося витратити години на те, щоб вирушити в бібліотеку, знайти потрібну книжку, вивчити її і вибрати інформацію, що цікавить, то зараз цей процес не завдає жодних незручностей. Увімкнув комп'ютер, ввів у пошуковій системі питання і через секунду на сторінці з'являться сотні посилань і величезна кількість інформації. Але відразу хочеться відзначити, що не варто вірити всьому, що пишуть в інтернеті і далеко не всі посилання наповнені достовірними фактами. Потрібно бути дуже уважним та перевіряти джерела. Дуже часто інформація в інтернеті є помилковою та її використання призводить до неприємних наслідків.

У сучасному світі все вже практично не зустріти будинку, де немає персонального комп'ютера. Комп'ютер нині грає роль підсилювача інтелектуальних можливостей людини й суспільства в цілому, а комунікаційні засоби, що використовують комп'ютери, служать для зв'язку та передачі. Поява та розвиток комп'ютерів та інформаційних – стало для всього людства практично нормою.

Багато людей вже не можуть уявити своє життя без персонального комп'ютера. Оскільки він великі можливості у спілкуванні або колективна розвага без відриву від зручного домашнього крісла. Все більшої популярності в цьому світі набувають інформаційні сайти, в яких можна знайти свою інформацію, яку ви можете зустріти на просторах онлайн території.

Тенденція до інформатизації спостерігається сьогодні майже скрізь: в освітніх установах; у засобах масової інформації; в урядових організаціях; у

різноманітних розважальних закладах; у промисловості; в оборонно-промисловому комплексі; у науково-дослідній діяльності; у засобах телекомунікацій; в будівництві; в освоєння водних, космічних, підземних просторів Землі та у багатьох інших галузях діяльності людства.

Не є винятком і різні автосалони, для яких інформаційний розвиток на сьогодні не менш важливий, ніж для комерційних організацій. За допомогою сучасних технологій автосалони можуть заявити про себе, спілкуватися з дилерами, ділитися досвідом у продажу продукції.

В історії обчислювальної техніки можна простежити дві основні області її використання: для виконання складних чисельних розрахунків та для зберігання та обробки великих обсягів інформації. Друга сфера застосування призвела до створення інформаційних систем.

На етапі розвитку суспільства одними з найважливіших напрямів є інформаційні технології. З кожним роком обсяг інформації незмінно збільшується, змушуючи витратити свою обробку дедалі більше тимчасових і трудових витрат. У зв'язку з цим дедалі важливішими стають сучасні автоматизовані інформаційні системи, які здатні за малі терміни обробляти вихідну інформацію та надавати її у зручному для користувача вигляді.

Інформаційні системи можуть принести величезну користь для корпорацій за рахунок автоматизації завдань, які раніше вирішувалися вручну. Якщо говорити коротко, то переваги інформаційних систем зводяться до таких ключових понять: швидше, краще та більше. Проте для того, щоб усвідомити користь інформаційних систем, необхідно мати можливість розробляти їх вчасно та з мінімальними витратами. Інакше кажучи, інформаційні системи повинні задовольняти інтересам бізнесу, бути легко модифікованими і недорогими. Погано спроектована система, зрештою, вимагає великих витрат та часу для її утримання та оновлення.

Ми живемо в епоху інтернету, тому щоб якийсь автосалон зібрав велику аудиторію треба розмістити її в інтернеті. У якому розповідається про саму гру та її можливості.

Для збільшення аудиторії автолюбителів передбачається впровадити веб-сайт, що містить веб-форум. На цей момент у Хакасії дуже бурхливо розвиваються веб-технології, для багатьох споживачів стає простіше знайти потрібну інформацію в інтернеті, ніж шукати її в газеті або чекати реклами по телебаченню. Також для всіх потенційних покупців буде доступна послуга веб-форуму, в якому користувачі зможуть обмінюватися своїм досвідом, порадами та різними порадами з автотематики.

Завдання, що реалізуються компаніями за допомогою Інтернету, умовно можна розбити на кілька типів:

- а). реклама та створення іміджу організації;
- б). маркетинг товарів;
- в). здійснення продажів за допомогою Інтернету;
- г). надання консультаційної допомоги споживачам;
- д). організація каналу мовлення ЗМІ.

Будь-який бізнес, який здійснюється за допомогою взаємодії з широкою аудиторією, потребує постійної реклами. Кілька років тому до звичайних засобів реклами додався новий, часом більш ефективний інструмент.

Починаючи з середини 90-х років, у всьому світі спостерігається зростання активності у галузі електронної торгівлі. Після великими компаніями, які виробляють комп'ютерне устаткування Інтернет стали виходити торговці традиційними товарами. З'явилася велика кількість книгарень, магазини компакт-дисків та відеокасет, винні магазини. Нині практично будь-які товари можна купити через Інтернет.

Представляючи інформацію про себе і продукти, що випускаються, фірма не тільки приваблює потенційних покупців, але і отримує можливість зворотного зв'язку з ними. Крім того, в Інтернеті простіше організувати адресну

рекламу, коли Web-сайт рекламується серед людей, які можуть бути потенційними покупцями.

Довгий час символом представницькості фірми була наявність Web-сайту, що саме по собі вважалося показником її гнучкості і хорошої здатності адаптуватися до умов бізнесу, що змінюються.

В даний час багато фірм, не шкодуючи коштів, розвивають свої сайти для надання кращого вигляду "особі" фірми у Всесвітній павутині. Web-сайти надають унікальну можливість для фірми ненав'язливо уявити великий обсяг інформації про себе, вироблені продукти або послуги. З іншого боку, за інтересом до того чи іншого товару зміну рейтингу відвідувань сайту можна судити про ефективність проведених рекламних акцій. Багато фірм, при випуску нових продуктів, спеціально створюють сайт з їх детальним описом, а потім через цей сайт отримують відгуки покупців та оперативно коригують свою діяльність щодо цих виробів.

Крім представлення опису товару, за допомогою Інтернету, можна здійснювати його продаж. Вже багато компаній мають свої електронні магазини.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

Для успішного створення працездатної, відмовостійкої та зручної інформаційної системи необхідне ретельне вивчення предметної галузі. Етап опису предметної області є найважливішим етапом під час створення будь-якої інформаційної системи. Від повноти і якості дослідження залежить доля створюваного програмного продукту, через те, що саме на цьому етапі виділяється коло автоматизованих завдань і закладаються вимоги до функціональності системи.

1.1. Аналіз предметної галузі

Електронний (віртуальний) магазин – це реалізоване в мережі Інтернет представництво шляхом створення Web-сервера для продажу товарів та послуг іншим користувачам мережі Інтернет. До нього цілком підходить визначення віртуального підприємства. Інакше кажучи, електронний магазин - це спільнота територіально роз'єднаних співробітників магазину (продавців, касирів) та покупців, які можуть спілкуватися та обмінюватися інформацією через електронні засоби зв'язку за повної (або мінімальної) відсутності особистого прямого контакту[5].

| | | | | | | | |
|------------------|--------------|--|--|--------------------------------|-------------|-------|---------|
| Кафедра КІТ (47) | | | | НАУ 21 26 69 000 ПЗ | | | |
| Розробив | Гончар В.О. | | | Дослідження предметної області | Літера | Аркуш | Аркушів |
| Керівник | Моденов Ю.Б. | | | | | 10 | 21 |
| Консультант | | | | | УС-212М 122 | | |
| Контролер | Райчев І.Е. | | | | | | |

Таблиця 1.1.

Порівняльна характеристика традиційної та електронної торгівлі

| | |
|---|---|
| Традиційний магазин | Електронний магазин |
| Торговий зал | Віртуальний торговий майданчик |
| Ходьба покупця по торговому залу та огляд товарів на полицях магазину | Перегляд покупцем сторінок сервера |
| Особистий контакт покупця із продавцем (консультація) | Консультація у продавця (при необхідності) по комп'ютерній мережі або по телефону |
| Вибір покупцем товару | Вибір покупцем товару |
| Замовлення товару | Замовлення товару через сервер |
| Виписка продавцем та вручення покупцю рахунку на оплату | Пересилання продавцем по комп'ютерних мережах покупцю рахунки на оплату |
| Оплата покупцем рахунку на товар у касі магазину готівкою | Оплата покупцем рахунку за якоюсь системою електронних платежів |

Електронна торгівля у віртуальному магазині ґрунтується на тій самій структурі, що й традиційна торгівля. Переведення традиційної торгівлі до мережі Інтернет робить її більш гнучкою, оскільки електронна торгівля, оперуючи цифровою інформацією комп'ютерних мережах, полегшує співробітництво людей[1].

Електронний магазин має доменну адресу. Як і будь-який Web-сервер, віртуальний магазин складається з цілого ряду гіпертекстових сторінок, найчастіше з мультимедійними елементами.

Вдалий Web-сайт - це надзвичайно ефективний інструмент торгівлі - він здатний захоплювати увагу аудиторії. Як і будь-який інший маркетинговий інструмент, заснований на принципі безпосереднього відгуку, насамперед він

повинен заінтригувати відвідувача, а потім спонукати його на певні дії. Однак багато хто ігнорує цю особливість головної сторінки, що часто призводить до того, що відвідувачі не затримуються на сайті надовго і залишають його, ледь зайшовши. Такі Web-сайти, нехай навіть містять іноді величезну кількість корисних порад і статей, практично ніколи не досягають передбачуваного рівня відвідуваності, не кажучи вже про продаж[13].

Зробивши кілька змін, простий Web-сайт може перетворитися на більш надійний і ефективний інструмент. Важливо пам'ятати, що день у день потенційних клієнтів обрушується потік інформації та різноманітних рекламних повідомлень, і що у плані завоювання їхньої уваги існує гранично жорстка конкуренція. Web-сайт, здатний привернути увагу і викликати цікавість, спонукає клієнтів не тільки переглянути сторінки, що залишилися, і здійснити покупки, але і знову відвідати його через деякий час, а також рекомендувати своїм друзям і знайомим[25].

Інтернет-магазин - це інтерактивний веб-сайт, що:

- Рекламує товар чи послугу в Інтернеті;
- Приймає замовлення для придбання;
- Пропонує користувачеві вибір варіанта розрахунку;
- Випишує рахунок на оплату, що служить одночасно підтвердженням

замовлення.

При цьому адміністратор магазину зобов'язаний:

- Організувати доставку товару;
- Проконтролювати розрахунки з покупцем за постачання.

Переваги Інтернет-магазину:

- Економія часу. Завдяки навігації та пошуковим системам можна швидко знайти потрібний товар;

- Доставка товарів додому. Користувачам Інтернет-магазинів більше немає потреби бігати по магазинах, тягати сумки та коробки. Все це швидко та легко роблять співробітники служби доставки;

- Можливість уникнути черг та суєти традиційних магазинів – проблема, яка викликала найбільше обурення у попередніх поколінь;
- Цілодобове обслуговування. Це так важливо і зручно сучасній людині, яка зайнята вдень і вночі, яка має ненормований робочий день і не встигає «до закриття магазину».

Класифікувати електронні магазини можна за різними критеріями. Найцікавішою класифікацією є класифікація за моделлю бізнесу:

- чисто онлайн-магазин;
- суміщення офлайн-бізнесу з онлайн-бізнесом (коли інтернет-магазин був створений на основі вже чинної реальної торгової структури).

Стосовно постачальників:

- Мають власний склад (наявність реальних товарних запасів);
- Працюють за договорами з постачальниками (відсутність значних власних запасів)[23].

Класифікація за товарним асортиментом – книги, аудіо, відеокасети, CD, DVD, комп'ютерна, побутова техніка, мобільні телефони.

Серед методів роздрібного продажу товарів у Мережі можна виділити:

- Інтернет-магазини (автоматичні магазини);
- Web-вітрини;
- Торгові автомати[23].

Internet-вітрина – це рекламний сервер. На вітрині викладають інформацію про товари, яку постійно оновлюють. Витрати її створення і адміністрування може бути досить низькими, а практична користь такої вітрини очевидна. Але це ще не торгівля. Потенційний покупець, завітавши до вітрини, повинен зателефонувати на фірму, оплатити товар, домовитися про доставку. Тому Internet-вітрина виправдана тоді, коли покупця треба познайомити зі складною продукцією, вивчення якої у торговому залі в нього піде дуже багато часу[17].

Internet-вітрина може бути розміщена будь-де - на власному сервері, на сервері провайдера, на сервері, що надає безкоштовні сторінки. Для роботи з вітриною достатньо мати підключення через телефонну лінію та мінімум навичок роботи з HTML[7].

Торговий автомат може виконувати функції вітрини, а й приймати замовлення і передавати їх менеджеру, тобто оформляти замовлення та виписувати рахунки на оплату без присутності покупця. Торговий автомат реально торгує і за співвідношенням витрат до результату найкращий для пілотних та тестових проектів з невеликим потоком покупців. Торговий автомат, як і Internet-вітрину, можна розмістити і своєму сервері, і сервері провайдера. Проте його створення та адміністрування вимагає навичок та певної кваліфікації[4].

Автоматичний магазин – ефективне та комплексне рішення у торговому бізнесі. Він не лише виписує рахунки, а й відстежує замовлення, приймає електронні платежі та формує заявки на доставку товарів покупцям. Тут завдання менеджера – контролювати роботу системи, складну в обслуговуванні.

Оскільки автоматичний магазин повинен мати постійний зв'язок з інформаційною системою компанії, то розміщувати його краще на корпоративному сервері в локальній мережі, або на віддаленому сервері з постійно діючим каналом зв'язку. Інтерактивні магазини можуть торгувати чим завгодно.

Отже, що бачить користувач, що зайшов у магазин?

По-перше, список товарів, що є на складі. Оскільки онлайн-вий «прилавок» зазвичай прив'язаний до системи автоматизації будь-якого підприємства, цей список містить ті ж вироби, що є у продажу й у звичайних (не віртуальних) магазинах. Вміст складу представляється зазвичай як ієрархічної деревоподібної структури, базовими елементами якої є групи товарів. Клацнувши мишею на групі, вона розгортається, відкриваючи список підгруп чи конкретних виробів певного типу. Іноді покупець може подивитися

картинку із зображенням товару та його характеристики, а також додати його до свого кошика.

Наповнивши кошик, клієнт віддає команду «Виконати замовлення» та вибирає зручну для нього форму оплати[3].

Якщо він здійснює покупку в магазині вперше, то його зазвичай просять вказати деякі відомості про себе - ім'я, телефон, адресу та ін. Корпоративний покупець повідомляє назву підприємства, номер розрахункового рахунку, ім'я та телефон контактної особи. На цьому етапі покупцеві надається певний ідентифікаційний код. Це робиться для того, щоб коли він зайде в магазин наступного разу, всю вказану інформацію можна буде не вводити - достатньо вказати свій код. Далі здійснюється розрахунок та безпосередня передача товару клієнту[29].

Існують різноманітні форми оплати: за готівку кур'єру при доставці, по безготівковому розрахунку (для організацій), банківським переказом, електронні платежі, оплата післяплатою або поштовим і телеграфним переказом.

Існують такі способи доставки замовлення: кур'єрськими службами, звичайною поштою або авіапоштою, самовивозом та інші способи доставки.

Типовий варіант інтернет-магазину складається з таких функціональних елементів:

- Каталог товарів;
- Пошукова система;
- Користувальницький кошик;
- Реєстраційна форма;
- Форма відправлення замовлення.

Каталог є складною і багаторівневою структурою даних, яка повинна простим і зрозумілим способом проводити впорядкування товарів. Найпростіше такий каталог у вигляді дерева об'єктів, верхній рівень якого складається зі списку розділів. Розділи можуть містити підрозділи чи

посилання конкретний товар тощо. Таке впорядкування просто необхідне для зручного та швидкого пошуку та замовлення товарів.

Пошукова система є обов'язковим елементом динамічного каталогу та реалізується на стороні сервера. Незважаючи на те, що каталог забезпечує впорядкування та групування даних, пошукова система дає користувачеві можливість швидкого пошуку інформації. Це особливо важливо в тому випадку, коли каталог являє собою досить розгалужену структуру даних з великою кількістю розділів, підрозділів і товарів, користувач погано представляє, в якому розділі може бути товар, що його цікавить і чи є він в каталозі взагалі. Пошукова система в деяких випадках дозволяє значно скоротити кількість переходів між сторінками каталогу для доступу до інформації, що цікавить[27].

Особливість реалізації пошуку в Інтернеті полягає в тому, що тут відбувається вибірка всіх записів, які відповідають умовам запиту (даний механізм пошуку називають пошуком із надлишком). У разі великої вибірки даних виведення результатів пошуку здійснюється посторінково для того, щоб відвідувачам не доводилося довго чекати завантаження всієї вибірки, яка може включати сотні, тисячі і більше записів. Як правило, відвідувачі не переглядають усі сторінки вибірки, обмежуючись двома чи трьома. Тому даний механізм пошуку у багатьох випадках працює надто повільно та неефективно. Однак він дозволяє здійснити вибірку однакових товарів від різних постачальників, порівняти їх параметри між собою та вибрати оптимальний варіант[24].

Користувальницький кошик є деяким масивом даних, який служить для зберігання замовленого користувачем товару.

Реєстраційна форма служить введення персональних даних користувачів. Надалі ця інформація використовується для їхньої ідентифікації між сеансами роботи з електронним магазином. Ця інформація може зберігатися як на стороні сервера, так і на стороні клієнта.

Форма відправлення замовлення служить для введення контактної інформації замовника та відправлення її та замовлення на електронну скриньку організації.

Електронний магазин можна реалізувати як за сервера, і за клієнта. У першому випадку використовують серверні сценарії, побудовані на таких технологіях як PHP, Perl, ASP, JSP, ColdFusion і т.п. У другому випадку – JavaScript (ActiveX, Java та ін. мають обмежене застосування). Вибір тієї чи іншої реалізації залежить від багатьох факторів, які впливають із визначених на етапі планування цілей та постановки задачі.

Для дослідження предметної області Web-сайту необхідно докладно розглянути структуру роботи автосалону.

Опції сайту.

Інформаційна система - така форма реклами продукції, за якої перегляд та вивчення товарів здійснюється через комп'ютерні мережі. При цьому як потенційні покупці товарів (або послуг) можуть виступати як приватні особи, так і організації.

Глобальна мережа Internet зробила інформаційну систему доступних фірм будь-якого масштабу. Якщо раніше організація електронного обміну доступних фірм помітних вкладень у комунікаційну інфраструктуру і була доступна лише великим компаніям, то використання Internet дозволяє сьогодні вступити до лав "електронних торговців" та невеликим фірмам. Інформаційна система WWW дає будь-якій компанії можливість залучати клієнтів з усього світу[33].

Інформаційній системі підпадають системи, орієнтовані на Інтернет - «інформаційні сайти», мова про які піде далі в курсовій роботі. Водночас процедури реклами, ініційованих інформацією з WWW, але які використовують для обміну даними факс, телефон та ін., можуть лише частково віднесені до класу Інформаційних систем.

Зазначимо також, що, всупереч тому, що WWW є технологічною базою інформаційної системи, у ряді систем використовуються й інші комунікаційні

можливості. Так, запити до продавця для уточнення параметрів товару або відгуків про нього можуть бути надіслані й через електронну пошту.

Інтернет – це глобальна комп'ютерна мережа, що охоплює весь світ. Сьогодні Інтернет має понад 400 мільйонів абонентів у більш ніж 150 країнах світу. Щомісяця розмір мережі збільшується на 7-10%. Інтернет утворює ніби ядро, що забезпечує зв'язок різних інформаційних мереж, що належать різним установам у всьому світі, одна з одною.

Інтернет складається з безлічі локальних та глобальних мереж, що належать різним компаніям та підприємствам, пов'язаних між собою різними лініями зв'язку. Інтернет можна уявити у вигляді мозаїки складеної з невеликих мереж різної величини, які активно взаємодіють одна з одною, пересилаючи файли, повідомлення тощо.

Причина, через яку мережа Інтернет стала настільки популярна, полягає у стандартизації процесів обміну даними (протоколів).

Своє «представництво», власну Web-сторінку в мережі має сьогодні практично кожна велика організація, фірма чи компанія. В Інтернет розташовані «електронні» варіанти багатьох тисяч газет та журналів, через Мережу ведуть мовлення сотні радіостанцій та телекомпаній. Важко знайти якусь область людської діяльності, яка не була б представлена в Інтернет у всій своїй повноті сотнями та тисячами «сторінок»[30].

Web-технологія повністю перевернула наші уявлення про роботу з інформацією та й з комп'ютером взагалі. Виявилось, що традиційні параметри розвитку обчислювальної техніки – продуктивність, пропускна здатність, ємність пристроїв, що запам'ятовують, – не враховували головного «вузького місця» системи – інтерфейсу з людиною. Застарілий механізм взаємодії людини з інформаційною системою стримував впровадження нових технологій та зменшував вигоду від їх застосування. І тільки коли інтерфейс між людиною і комп'ютером був спрощений до природності сприйняття звичайною людиною, був безпрецедентний вибух інтересу до можливостей обчислювальної техніки.

З розвитком технологій гіпертекстової розмітки в Інтернеті стало з'являтися все більше сайтів, тематика яких була абсолютно різною - від сайтів великих компаній, що оповідають про успіхи компанії та її провали, до сайтів маленьких фірм, що пропонують відвідати їхні офіси в межах одного міста.

Мережа дає будь-якій людині практично безоплатну можливість сповістити багатомільйонну аудиторію про запропоновані їм послуги чи продукцію. Інтернет зрівнює приватних осіб, фірми середньої руки та великі корпорації: всі мають однакові можливості для залучення покупців. Не треба платити тисячі і навіть мільйони доларів за миті реклами на телебаченні, не треба купувати смуги в газетах – ваша сторінка в Інтернеті функціонуватиме цілодобово, без перерв.

Все більш популярною стає електронна торгівля, що дозволяє користувачеві здійснити покупку практично будь-якого товару у будь-якій точці планети. По Інтернету ви можете замовити та отримати нові програмні продукти, надіслати букет квітів коханій дівчині та навіть придбати автомобіль. А також дізнатися про останні результати торгів на біржах усього світу, дізнатися про курс акцій тієї чи іншої компанії та перевірити з ними угоду. Для великих фірм та корпорацій Мережа стала ідеальним середовищем для проведення різноманітних операцій та розрахунків, а також торгівлі за схемою business-to-business, нарад у реальному часі. Втім, заробити на Мережі може не лише велика фірма, а й практично будь-яка людина, яка створила свою сторінку.

Але організації, діяльність яких пов'язана безпосередньо з програмуванням і дизайном, що неспроможні створити собі Web-сайт самостійно. Для того, щоб успішно конкурувати та заробляти у всесвітній мережі компанії вдаються до послуг Web-майстрів та веб-студій. Але чи буде це під силу малій фірмі, чи можна бути впевненим у надійності та працездатності зроблених інтернет-сторінок, у якому стані взагалі знаходиться ринок даних послуг – на ці та багато інших питань допоможе відповісти маркетингове дослідження ринку послуг веб-розробки[31].

Веб-сайт – це набір документів та файлів, які лежать на якомусь сервері (комп'ютері) та об'єднані одним доменним ім'ям. Ім'я домену, це назва веб-сайту, яка використовується для його пошуку у мережі. Наприклад, "mail.ru" є доменним ім'ям, яке поєднує сторінки, що лежать на сервері.

Сервером може бути звичайний комп'ютер, на якому встановлені певні програми, що дозволяють працювати на веб-сайті. Браузер надсилає певний запит, який є доменним ім'ям веб-сайту, сервер його приймає і відправляє назад деякі документи. Це працює як мережне копіювання файлів, тільки через протоколу http. Усі документи веб-сайту, які ми бачимо у браузері, написані на мовою HTML.

HTML (HyperText Markup Language) це мова, яка є стандартом для розробки документів, які відобразатимуться в інтернет-браузерах. Ключовий зміст мови HTML у розмітці тексту, для чого використовуються спеціальні команди, звані теги, графічно вони відображаються у кутових дужках. HTML-документи хороші тим, що для них перегляду підходить практично будь-який браузер. Документ, написаний на мовою HTML, буде правильно відобразитися на різних пристроях, мають різну роздільну здатність екрана та операційну систему[12].

Практично всі документи містять стандартні елементи, такі, як титульний напис, заголовки чи списки. За допомогою тегів HTML можна позначати ці елементи, забезпечуючи веб-браузери мінімальною інформацією для відображення даних елементів, зберігаючи в загальному загальну структуру та інформаційну повноту документів.

Браузер – це вікно, через яке ми перегледаємо веб-сайти. Тобто все сторінки мають бути написані мовою HTML.

Створити документи, написані мовою HTML, які стануть основою вашого веб-сайту можна кількома способами.

Основним способом є текстовий набір HTML-коду редактор. Це досить трудомісткий процес, який має на увазі глибоке знання принципів побудови

документів у HTML, вміння працювати з каскадними таблицями стилів CSS, а також як мінімум з JavaScript.

Другим способом використання CMS. Це двигун сайту, який дозволяє користувачеві майже автоматично створювати веб-сайти, має на увазі менший набір знань для створення і потрібно набагато менший час на розробку. Також при використанні CMS набагато простіше здійснювати подальше обслуговування веб-сайту, додавання чи видалення нову інформацію.

Також є можливість створювати веб-сайти за допомогою спеціальних редакторів, які об'єднуються під загальною назвою WYSIWYG-«What You See Is What You Get», який можна перекласти як «що ти бачиш, то ти і отримуєш». Створення веб-сайтів у цих програмах нагадує роботу у графічних та текстових редакторах, але на виході ми отримуємо не зображення, а документ мовою HTML. Цей метод дозволяє достатньо швидко отримати документ із досить складним оформленням[26].

Розглянемо переваги та недоліки цих методів. Спочатку здається, що переваги CMS або редакторів WYSIWYG незаперечні, але це не так. Веб-сайти, написані вручну, хоч і потребують більшого часу на створення, зате виходять якісніше. Позначимо переваги веб-сайтів, написаних вручну: Створений таким чином веб-сайт набагато складніше зламати, оскільки немає MySql бази даних і відсутній файл конфігурації, а саме з За допомогою цих файлів в першу чергу хакери намагаються зламати веб-сайт.

Веб-сайти, написані на HTML, якщо порівнювати їх із CMS системами відображаються набагато швидше. Так виходить тому, що самі файли виходять набагато компактнішими і не треба робити запити в MySql бази даних.

Швидкість відображення веб-сайту позитивно впливає на позицію видачі під час пошукових запитів. За однакових умов веб-сайт, написаний на HTML буде вищим. Веб-сайт займає менше місця, що дозволяє заощаджувати на тарифах хостингу. HTML сайт не міститиме продубльованих сторінок, що позитивно відображається на результатах пошуку. Для опису зовнішнього вигляду елементів застосовується каскадні таблиці стилів CSS.

CSS досить елементарна формальна мова, яку вигадали для опису зовнішнього вигляду документів. Це говорить про те, що він досить простий і складається з самотніх примітивних конструкцій, які не так складні вивчення. Найскладніше не синтаксис, не правила написання конструкцій, а величезна кількість CSS властивостей для запам'ятовування, які виконують різноманітні завдання. Благо, всі правила англійські відповідним смисловим навантаженням.

Простий переклад на нашу мову дає поняття про те, що це правило робить і навпаки – при перекладі того, що ми хочемо добитися певною властивістю на англійську мову, велика ймовірність того, що ми отримаємо правильну властивість. Це значно спрощує запам'ятовування CSS правил інтуїтивному рівні. Наприклад, якщо потрібно задати фоновий колір достатньо зробити переклад на англійську, внаслідок чого отримуємо `background-color` (окремі слова в CSS пишуться через дефіс)[25].

1.2. Аналіз платіжних систем

Оскільки програмний додаток є електронним магазином, неможливо обминути тему оплати товару в мережі Інтернет. Спочатку розглянемо існуючі системи оплати в мережі Інтернет.

Традиційні методи оплати, включаючи готівку, банківські перекази, чеки, пластикові картки, винайдені задовго до електронної комерції. Тому немає нічого дивного в тому, що вони не повністю відповідають її потребам. Безумовно, купуючи звичайного «фізичного» товару гроші можна взяти з покупця під час доставки. Але якщо купується «цифровий» товар чи інформація, обов'язково має бути спосіб сплатити покупку прямо «на місці», тобто на сайті продавця. Ось чому в електронній комерції надзвичайно гостро стоїть питання розробки універсального способу оплати покупок в Інтернет, який дозволив би робити дешеві та безпечні платежі в режимі реального часу.

До способів оплати покупок в Інтернет, які проводяться електронним шляхом, можна віднести такі:

- оплата з рахунку мобільного телефону (m-commerce);
- оплата за картами передоплати (скретч-карт);
- оплата електронними грошима;
- оплата платіжними картками.
- оплата з рахунку мобільного телефону (m-commerce).

Мобільна комерція це використання мобільних портативних пристроїв для спілкування, розваги, отримання та передачі інформації, здійснення транзакцій через громадські та приватні мережі.

Сьогодні на ринку з'явилася чудова можливість оплачувати послуги чи товари різних компаній через мобільний телефон. Ви можете робити покупки в Інтернет-магазинах, сплачувати комунальні платежі і навіть переказувати гроші за банківськими реквізитами. Незалежно від того, що і як Ви сплачуєте, процес оплати займає всього кілька хвилин, а сам платіж відбувається майже миттєво.

Деякі мобільні компанії надають цю послугу безкоштовно як частину звичайного пакета користувача мобільного телефону. Гроші, які будуть використані власником мобільного телефону для придбання товарів або послуг, можуть бути зняті з рахунку власника телефону або зі спеціальної картки, яку потрібно спочатку купити, а потім активувати зі свого телефону.

Оплата послуги або товару відбувається за допомогою посилки секретного повідомлення (SMS), що включає ланцюжок інформації типу номер рахунку власника телефону/сума до оплати або кодівий номер товару або послуги/ номер рахунку продавця[33].

Якщо розмір платежу, що здійснюється, перевищує залишок коштів на поточній картці, необхідно активувати нову картку. При цьому невикористані залишки коштів з попередніх карток додаються до номіналу активованої картки.

Важливо, що для того, щоб Ви могли використовувати цю функцію, Ваш телефон повинен бути оснащений функцією WAP або деяким власним мікробраузером. Мобільна комерція робить користувача ще більш незалежним, не прив'язаним до стаціонарних пристроїв, надаючи всі перераховані вище можливості за наявності одного тільки мобільного телефону або кишенькового комп'ютера. Це дуже важливо для ділової людини: часто залежить від миттєво прийнятого рішення, і цьому не повинні перешкоджати такі фактори, як неможливість швидкого оформлення угоди або відсутність доступу до інформаційних каналів.

Оплата за картами передоплати (скретч-карт).

Скретч-карти є картами передоплати, оскільки, купуючи їх, ви робите передоплату сервісу або товару, який збираєтеся замовити. Свою назву вони отримали від англійського слова "scratch", що означає дряпати. Саме наявність особливого захисного шару приховує буквено-цифровий набір є відмінною особливістю карт передоплати. Дуже важливо відзначити, що кожна картка передоплати має свій унікальний серійний номер, який використовуватиметься для оплати послуг.

Карты передоплати випускаються різними фірмами, бувають різного номіналу та різного призначення. Як правило, скретч-карта має обмежений термін дії, який може змінюватись в залежності від номінальної вартості картки або інших умов. Сама карта передоплати являє собою пластикову картку з розміщеною на ній інформацією щодо фірми виробника, сфери її застосування та терміну її дії[16].

Карты передоплати набули досить широкого поширення, оскільки є зручним способом оплати послуг та покупок. Їх активно використовують компанії, що займаються стільниковим зв'язком, наданням послуг Інтернет-провайдерів тощо.

Оплата електронними грошима

Цифрові гроші є основою платіжної системи Інтернету. Вони можуть бути класифіковані за видами організації їхнього функціонування, за рівнем безпеки, а також за способом розрахунку.

Цифрову готівку можна уявити як файли-жетони, що замінюють готівку. Продавці та покупці можуть вільно обмінюватися цими "монетами" через мережу, оплачуючи ними товари та послуги. Для зазначеної мети учасники системи встановлюють у себе на комп'ютерах особливу програму - "електронний гаманець", який забезпечує облік та передачу жетонів, а також перевірку їхньої справжності. Цифрові гроші можуть необмежено довго звертатися в мережі, але можуть у будь-який час бути обмінені на справжні гроші в організаторів системи або в банках, що беруть участь у ній.

Різновидом електронних грошей є системи цифрових готівки, що базуються на використанні технології smart-card.

Сучасна смарт-карта - це маленький комп'ютер зі своїм процесором, пам'яттю, програмним забезпеченням та системою введення/виведення інформації. У чіпі такої картки зберігається не що інше, як електронні гроші в описаному вище сенсі.

Поки що смарт-карта використовується як звичайна дебетова карта (називається електронним гаманцем), в яку вносяться записи про списання грошей, або просто інформація про клієнта. Готівкові цифрові гроші на базі смарт-карток не тільки можуть забезпечити необхідний рівень конфіденційності та анонімності, але й не вимагають зв'язку з центром для підтвердження оплати на відміну від подібних систем на базі персонального комп'ютера. У зв'язку з цим вартість транзакції прагнуть до нуля, за винятком поповнення картки або переведення в готівку грошей. Єдина незручність полягає в тому, що для переведення електронних грошей з картки на комп'ютер або назад необхідно особливий пристрій для читання карток, карт-рідер, що приєднується до комп'ютера.

Оплата платіжними картками

Кредитна картка - іменний платіжно-розрахунковий документ у вигляді пластикової картки, що видається банком своїм вкладникам для безготівкової оплати ними товарів та послуг у роздрібній торговельній мережі, з комп'ютерними пристроями, що передають запит на оплату товару до банку.

Кредитна картка засвідчує наявність у її власника поточного рахунку у банку. У різних країнах використовують кредитні картки американських кредитно-фінансових груп типу Visa, MasterCard, American Express і т.д.

Дебетові картки можуть використовуватися при оплаті товарів та послуг через Інтернет у режимі он-лайн так само, як при отриманні готівки в банкоматі: для здійснення платежу клієнт повинен запровадити номер картки та PIN-код.

Віртуальні карти

Оскільки використання класичної кредитної картки для розрахунків в Інтернеті є небезпечним, були розроблені віртуальні картки типу VISA Ec@rd та Virtual MasterCard, які призначені виключно для розрахунків в Інтернеті. VISA Ec@rd розроблена за технологією міжнародної платіжної системи VISA Int. Virtual MasterCard розроблена за технологією міжнародної платіжної системи MasterCard Int[16].

Випуском подібних карток сьогодні займаються різні банки своєчасно оцінивши їх переваги та потенціал. VISA Ec@rd та Virtual MasterCard застосовується виключно для оплати через Інтернет товарів та послуг в електронних магазинах по всьому світу, а також послуг операторів мобільного зв'язку, провайдерів Інтернету, туристичних фірм та готелів.

До покупок, сплачених не електронним шляхом, можна віднести:

- оплата кур'єру готівкою;
- передоплата за безготівковим розрахунком;
- оплата післяплатою.
- оплата кур'єру готівкою.

Один із найпростіших способів оплати це оплата кур'єру. Ви просто вручаєте гроші кур'єру та розписуєтеся в отриманні замовлення. На жаль, цей спосіб можливий не скрізь.

Після вибору товару в інтернет-магазині покупець розміщує замовлення, в якому вказує адресу для доставки (додому, в офіс, або ще кудись). Кур'єр привозить товари за вказаною адресою та отримує гроші.

1. Переваги цього, напевно, можна перераховувати нескінченно.

Недоліки також очевидні:

- завжди приємно бачити у себе вдома (або в офісі) незнайому людину (кур'єра), спілкуватися з ним;
- найчастіше очікування кур'єра перетворюється на тортур, і відбиває будь-яке бажання щось купувати.

Передоплата за безготівковим розрахунком

Передоплата за безготівковим розрахунком можлива для організацій. Під час оформлення замовлення вказується назва організації. При цьому можна заповнити та роздрукувати рахунок. Щойно гроші прийдуть на рахунок, магазини надсилають замовлення.

Оплата післяплатою

Оплачувана покупка при отриманні на поштовому відділенні називається оплатою післяплатою. При оплаті післяплатою, пошта стягує 8% від вартості замовлення.

Порядок оплати такий:

- адреса надсилається замовлення, або заповнюється певна форма оплати запропонована інтернет-магазином;
- вказуються свої П.І.Б., повна адреса з індексом, контактний телефон та e-mail;
- При отриманні поштового повідомлення, перераховується поштові витрати та вартість замовлення на зазначені банківські реквізити;

- далі за отриманим у банку квиток поштовими працівниками видається замовлення.

На початковому етапі в проекті інтернет-магазину будуть реалізовані способи оплати як передплата за безготівковим розрахунком (зручно для корпоративних клієнтів) та оплата кур'єру готівкою (зазвичай використовується при невеликих сумах, а також приватними особами).

1.3. Постановка задачі

Як було зазначено вище, кількість прикладних додатків в Інтернеті стрімко збільшуються і особливо актуальні проектні рішення з використанням БД, що дозволяють вирішувати багато завдань.

З урахуванням великої кількості ресурсів, що розташовуються в інтернет, сучасні Web-вузли зазвичай використовують мережеві СУБД. Застосування таких СУБД дозволяє швидко та ефективно обробляти запити користувача та надійно зберігати дані. У дипломному роботі розробляється Web-вузол з прикладу бази даних з «електронної комерції». На даний час Інтернет-магазини є найпоширенішою послугою в мережі Інтернет.

Виходячи з характеристик різних технологій, більш зручною, простою, ефективною, а також безкоштовною була обрана технологія PHP. До конкретних завдань належать:

- розробка структури Web-вузла;
- вибір програмного інструментарію;
- розробка зручного і зрозумілого інтерфейсу користувача;
- розробка структури та форм запитів;
- розробка раціональної навігаційної структури системи (Web-вузла).

- розроблений Web-проект повинен бути оптимізований під використанням у реальній мережі (мати мінімально можливий обсяг запитуваних Web-сторінок).

До загальних вимог слід віднести обсяг інтерактивних та графічних елементів, що має бути мінімізований. Інтернет-магазин повинен мати систему захисту від несанкціонованого доступу. Система має бути орієнтована на подальший розвиток.

Основною вимогою розробки інформаційного забезпечення є вибір СУБД (Система Управління Базою Даних) для зберігання інформації про товари. Як така СУБД була обрана MySQL, так як дана СУБД має клієнт - серверну архітектуру, функціонує в операційних системах сімейства Unix, займає невелику кількість машинної пам'яті. Підтримка сервера MySQL автоматично включається до постачання PHP[14].

Функції сайту організації

Цей сайт є інтерактивною системою, яка забезпечує наступні функції:

- Інформативна функція: сайт дозволяє оперативно порівнювати необхідні товари та забезпечує простоту пошуку.
- Практична функція: web-сайт також прискорює процес написання повідомлення та пред'явлення його на виконання адміністратору.

Процес написання повідомлення

Для будь-якого користувача процес написання дуже простий та зрозумілий. Зайшовши на форум або гостьову книгу, користувач залишає повідомлення, яке публікується або обробляється адміністратором. Вся інформація, внесена користувачем, надходить у книгу, обробка її ведеться оперативно і коректно.

ВИСНОВКИ ДО РОЗДІЛУ 1

Для успішного створення працездатної, відмовостійкої та зручної інформаційної системи необхідне ретельне вивчення предметної галузі. Етап опису предметної області є найважливішим етапом під час створення будь-якої інформаційної системи. Від повноти і якості дослідження залежить доля створюваного програмного продукту, через те, що саме на цьому етапі виділяється коло автоматизованих завдань і закладаються вимоги до функціональності системи.

У роботі над даним розділом були поставлені задачі на виконання дипломної роботи. Проаналізовано платіжні системи та скомпоновано ідеальний варіант використання системи оплат на веб порталі автосалону.

РОЗДІЛ 2. ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Етапи створення веб-сайту

1. Створюється модель чи прототип веб-сайту. На цьому етапі приблизно накидається зовнішній вигляд веб-сайту та загальна структура веб-сайту, визначається які елементи та в якому порядку він буде утримувати. Для цього можуть бути використані різні програми та онлайн сервіси, але також можна просто відобразити на аркуші паперу.

2. Створення дизайну веб-сайту. Як правило, макет, що відображає дизайн веб-сайту, малюється в графічних редакторів, таких як Adobe Photoshop або Gimp або Adobe Experience. Це є одним із головних етапів створення веб-сайту. Саме на цьому етапі остаточно формується зовнішній вигляд веб-сайту, додається вся графіка, уточнюються та промальовуються всі деталі.

3. Верстка сторінок у HTML. Макет веб-сайту, намальований у графічному редакторі, розбивається на окремі графічні елементи та описуються на мові HTML, також при верстці можуть бути використані такі технології, як CSS та javascript, якщо необхідне якість динамічне підвантаження або анімація.

4. Розробляється програмна частина або використовується двигун.

5. Веб-сайт розміщується на хостингу. Підготовлені HTML сторінки копіюються на сервер, їм присвоюється певне доменне ім'я та отримують публічний доступ [14].

Адміністративна частина сайту представлена у вигляді системи керування вмістом.

| | | | | | | | |
|------------------|--------------|--|--|--------------------------------|-------------|-------|---------|
| Кафедра КІТ (47) | | | | НАУ 21 26 69 000 ПЗ | | | |
| Розробив | Гончар В.О. | | | Вибір програмного забезпечення | Літера | Аркуш | Аркушів |
| Керівник | Моденов Ю.Б. | | | | | 31 | 30 |
| Консультант | | | | | УС-212М 122 | | |
| Контролер | Райчев І.Е. | | | | | | |

Система управління вмістом (англ. Content management system, CMS) - програма, використовувана для управління вмістом чогось (зазвичай цей вміст розглядається як неструктуровані дані предметного завдання на противагу структурованим даним, які зазвичай знаходяться під управлінням СУБД). Зазвичай такі системи використовуються для зберігання та публікації великої кількості документів, зображень, музики чи відео.

Приватним випадком таких систем є системи управління сайтами. Подібні CMS дозволяє керувати текстовим та графічним наповненням веб-сайту, надаючи користувачеві зручні інструменти зберігання та публікації інформації[6].

Зараз існує безліч готових систем управління вмістом сайту, у тому числі безкоштовних. Їх можна поділити на три типи, за способом роботи:

Генерація сторінок на запит. Системи такого типу працюють на основі зв'язки «Модуль редагування → База даних → Модуль представлення». Модуль представлення генерує сторінку із вмістом при запиті на нього, на основі інформації з бази даних. Інформація у базі даних змінюється за допомогою модуля редагування. Сторінки знову створюються сервером при кожному запиті, а це створює навантаження на системні ресурси. Навантаження може бути багаторазово знижено при використанні засобів кешування, які є в сучасних web-серверах.

Генерація сторінок під час редагування. Системи цього типу є програмою для редагування сторінок, які при внесенні змін до змісту сайту створюють набір статичних сторінок. При такому способі жертвується інтерактивність між відвідувачем та вмістом сайту.

Змішаний тип. Як відомо з назви, поєднує у собі переваги перших двох. Може бути реалізований шляхом кешування - модуль вистави генерує сторінку один раз, надалі вона в рази швидше підвантажується з кеша. Кеш може оновлюватися як автоматично, через деякий час або при внесенні змін до певних розділів сайту, так і вручну за командою адміністратора. Інший підхід -

збереження певних інформаційних блоків на етапі редагування сайту та складання сторінки з цих блоків при запиті відповідної сторінки користувачем.

Обґрунтування вибору CMS (Content Management System) необхідна розвинена гіпертекстова структура - HTML створений саме для цього; необхідно забезпечити гнучку структуру управління матеріалом - керувати HTML-змістом зручно за допомогою PHP + MySQL. Саме це зв'язування дозволяє генерувати динамічні HTML-сторінки; необхідно реалізувати варіативність представлення матеріалу – у разі використання HTML+CSS більш ніж обґрунтовано. Саме це комбінація коштів дозволяє створювати мультимедійні документи; необхідно забезпечити підтримку внутрішнього зворотного зв'язку веб-сайту[4].

2.1. Розмітка сторінок - HTML

Всесвітня павутина (World Wide Web) спровокувала революцію в інформатиці, надавши будь-якому користувачеві можливість публікації HTML-документів. Донедавна інформація у цих документах була здебільшого статичною, що потребувало реакції сервера на дії користувача. З введенням динамічного HTML парадигма Web змістилася від взаємодії із сервером у бік створення інтерактивних Web-вузлів та Web-додатків. Оскільки динамічний HTML забезпечує можливість взаємодії HTML-документів з користувачем і повної їх зміни на клієнтському комп'ютері, можливо створювати Web-додатки з багатими можливостями. У документі HTML/SGML теги визначають структуру змісту документа. Традиційний SGML-документ має три різні компоненти: структура, стиль і зміст. З введенням динамічного HTML було додано четвертий компонент: модель поведінки. Термін модель поведінки (behavior) визначає взаємодію між користувачем та HTML-сторінкою. Структура представлена низкою сімейств елементів; стиль представлений для кожного елемента та у всіх сімействах таблиць стилів. Зміст представлений за

допомогою всіх елементів та об'єкта `TextRange`. Сценарії (скрипти) маніпулюють структурою, стилем та змістом у відповідь на події для створення моделі поведінки документа.

Очевидно, що мова HTML збереже свої позиції як основного засобу для розмітки WWW-документів, завдяки своїй простоті та винятковій корисності.

Вже досить багато часу витрачено, щоб навчитися за допомогою HTML і браузера виводити на екран тексти.

Але це просто тексти. Використовуючи потужний текстовий процесор, такий, як Microsoft Word, зверстати оголошення, лист брошуру та невелику книгу можна набагато швидше, і результат буде кращої якості.

При цьому не треба писати програми, вникати у всі тонкощі численних тегів та їх атрибутів. Працюючи у Word, практично не треба думати ні про що, крім змісту власного твору. Різноманітний оформлювальний інструментарій має інтуїтивно-зрозумілий інтерфейс та численну бібліотеку шрифтів, ліній, значків, рамок, орнаментів, фігур, картинок та інших корисних "штучок", які роблять текст на екрані та папері приємним для ока. За умови, звичайно, що ви не позбавлені смаку та почуття міри.

То чому ж у всьому світі вибухнув справжній бум HTML-програмування?

Головних причин популярності HTML є три. Ось вони в порядку зростання ваги.

HTML-програмування дуже просте. У ньому немає традиційних алгоритмічних структур, таких як розвилка, цикл, процедура. Воно лінійне у своїй основі.

(Відступом від цієї тези є посилання. З їхньою допомогою можна організувати розгалуження та процедурний виклик. Про це буде розказано нижче. Проте, ці засоби надто примітивні для серйозного зіставлення з аналогічними структурами мов типу Сі або Паскаль.)

Багато авторів підручників навіть соромляться називати HTML-тексти програмами. Роботу HTML-проектувальника називають розміткою тексту.

Насправді, у цій назві міститься методична помилка. Адже вона (назва) передбачає такий порядок роботи:

береться звичайний текст і розмічається, тобто на нього накладається сітка з тегів так, щоб браузер зміг показати рядки тексту на екрані.

Однак, хороші гіпертекстові документи при такому підході отримати набагато складніше, ніж у випадку, коли структура HTML-документа планується ще до написання текстів. Адже гіпертекст влаштований принципово інакше (див. третю причину).

Переносність.

Якщо ви працюєте в Word'e, ви створюєте документ для власного споживання, у вас немає проблем. Немає проблем і тоді, коли ви надсилаєте свою роботу у світ як друковану копію. Якщо ваш партнер, видавець або колега просить надіслати електронну версію - починаються проблеми.

Щоб документ з'явився на екрані у абонента, необхідно, щоб його комп'ютер та операційна система (платформа) дозволяли запустити ваш улюблений редактор.

Для того, щоб документ завантажився в Word, дуже бажано збіг версій цього продукту у вас та вашого колеги.

Щоб текст на екрані не виглядав як давньонорвезький манускрипт, необхідно, щоб ваш партнер мав на своєму комп'ютері шрифти, які були використані при створенні вашого тексту.

Але навіть якщо всі ці умови виконані, все одно мало шансів на те, що ваш товариш побачить текст неспотвореним (картинки вилазять з рамок, межі сторінок пливають текстом,...). Це відбувається тому, що ваш Word та Word вашого партнера налаштовані по-різному.

Після того, як по телефону (або електронній пошті) редактори будуть налаштовані однаково, текст все одно може виглядати по-різному вже через причини, відомі лише фірмі Microsoft.

Все, що говорилося, називається непереносимістю продукту. Не в тому сенсі, що продукт когось не любить, а в тому, що різним користувачам неможливо або вкрай важко відобразити у себе те, що зроблено в іншому місці.

HTML-документ є переносним продуктом. Це означає, що авторська праця буде легко доступна величезному числу користувачів, незалежно від марки комп'ютера та типу операційної системи. Саме ця властивість HTML-документів, а також відносно малий розмір, дозволили з успіхом використовувати HTML-технологію для підготовки WWW-сторінок в Інтернеті.

Перенесення HTML-документа досягається за рахунок того, що пересилається не екранне зображення, а програма. А зображення будує браузер, виконуючи команди цієї програми. Щоправда, і тут не все гладко. Різні браузери можуть працювати трохи по-різному.

HTML-документ – це гіпертекст[4].

Традиційний текст має лінійну структуру. І хоча користувач може читати будь-які його сторінки та рядки, гортаючи документ на екрані або паперовій книзі, автор припускає, що текст читається по порядку. Спочатку перша сторінка, потім друга і таке інше.

2.2. JavaScript

Мова програмування, як і будь-яка інша мова (природна, наприклад), призначена для комунікації, тобто зв'язку між тим, хто говорить і слухає. У програмуванні мовцем є програміст, а слухачем — інтерпретатор мови, деяка комп'ютерна програма, яка розуміє цю мову і виконує дії відповідно до того, що вона зрозуміла. Був час, коли вважалося, що для полегшення спілкування з комп'ютером необхідно створити мову досить близьку до природного. Ця ідея, зрештою, не витримала випробувань часом, хоча породила кілька прекрасних

мов програмування. Часто буває, що побічні ефекти якоїсь діяльності перевершують очікування.

При генерації сторінок у Web виникає дилема, пов'язана з архітектурою "клієнт-сервер". Сторінки можна генерувати як за клієнта, і за сервера. У 1995 році фахівці компанії Netscape створили механізм управління сторінками на стороні клієнта, розробивши мову програмування JavaScript.

Таким чином, JavaScript – це мова керування сценаріями перегляду гіпертекстових сторінок Web на стороні клієнта. Якщо бути точнішим, то JavaScript – це не лише мова програмування на стороні клієнта. Liveware, прабатько JavaScript, є засобом підстановок на стороні сервера Netscape. Проте найбільшу популярність JavaScript забезпечило програмування за клієнта.

Основна ідея JavaScript полягає у можливості зміни значень атрибутів HTML-контейнерів та властивостей середовища відображення у процесі перегляду HTML-сторінки користувачем. При цьому перезавантаження сторінки не відбувається.

Насправді це виявляється у тому, що можна, наприклад, змінити колір фону сторінки чи інтегровану у документ картинку, відкрити нове вікно чи попередження.

Мова програмування, що інтерпретується, є однією з реалізацій мови ECMAScript і заснований на концепції прототипів, що прийшла з мови Self [3].

В даний час JavaScript використовується в основному для створення сценаріїв поведінки браузера, що вбудовуються в веб-сторінки, але також знаходить застосування як мову скрипта доступу до об'єктів додатків. Деякі програми мають вбудований інтерпретатор JavaScript, що дозволяє розширювати їх можливості без зміни програми.

Мова програмування JavaScript розроблена фірмою Netscape у співпраці з Sun Microsystems та призначена для створення інтерактивних HTML-документів. Основні області використання JavaScript такі:

а) створення динамічних сторінок, тобто сторінок, вміст яких може змінюватись після завантаження документа;

б) перевірка правильності заповнення користувачем форм перед пересиланням їх на сервер;

в) вирішення "локальних" завдань за допомогою сценаріїв та деякі інші сфери. дозволяє створювати додатки, що виконуються як на стороні клієнта, так і на стороні сервера. При розробці програм обох типів використовується так зване ядро, в якому містяться визначення стандартних об'єктів. Клієнтські програми виконуються браузером на машині користувача. Програма (сценарій) мовою JavaScript обробляється вбудованим в браузер інтерпретатором. Треба прагнути того, щоб написані сценарії коректно виконувались у будь-якому браузері.

Код JavaScript включається до HTML-коду сторінки і виконується інтерпретатором, вбудованим у браузер (таблиця 2.1)[3].

Таблиця 2.1.

Версії мови JavaScript та відповідні їм браузери

| Версія мови | Версія Netscape Navigator | Версія Internet Explorer |
|------------------------------|---------------------------|--------------------------|
| JavaScript 1.0 (JScript 1.0) | 2.x | 3.x |
| JavaScript 1.1 | 3.x | - |
| JavaScript 1.2 (JScript 3.0) | 4.0 | 4.x |
| JavaScript 1.3 | 4.06 | - |
| JavaScript 1.4 (JScript 5.0) | - | 5.x |
| JavaScript 1.5 | 6.0 | - |

Ядро JavaScript містить набір основних об'єктів, таких як Array, Date та Math, та основний набір елементів мови, таких як операції, керуючі структури та оператори. Ядро JavaScript може бути розширене для різних цілей шляхом доповнення його до нових об'єктів; наприклад:

а) клієнтський JavaScript розширює ядро мови, додаючи об'єкти керування браузером (Navigator або інший аналогічний web-браузер) та

Document Object Model/Об'єктну Модель Документу (DOM). Наприклад, клієнтські розширення дозволяють застосунку розміщувати елементи на HTML-формі і реагувати на події користувача, такі як клацання мишею, введення даних у форму або навігація по сторінках;

б) серверний JavaScript розширює ядро мови, додаючи об'єкти, що стосуються запуску JavaScript на сервері. Наприклад, розширення сервера дозволяють застосунку взаємодіяти з реляційною базою даних, зберігаючи інформацію між викликами програми, або виконувати маніпуляції з файлами на сервері.

2.3. Server pages

Page - інтерфейс доступу до баз даних або програм мовою Java; дозволяє формувати динамічні сторінки на Web-сервері. Pages (JSP) дозволяють відокремити динамічну частину Web-сторінок від статичного HTML. Спочатку, як завжди, пишеться звичайний код HTML, використовується для цього будь-яка програма для створення Web-сторінок. Потім динамічну частину коду полягає у спеціальні таги, більшість яких починаються з "<%>" і завершуються "%>".

Файлу дається розширення ".jsp" і він розміщується там, де мають розміщуватися звичайні Web-сторінки. Хоча те, що написано більше схоже на звичайний HTML файл ніж на сервлет, просто за кадром JSP сторінка перетворюється на звичайний сервлет зі статичним HTML який просто спрямовується в потік виводу, пов'язаний з методом сервлета service. Зазвичай це відбувається при першому запиті сторінки, і розробники можуть відразу після установки самі виконати цей запит, якщо хочуть, щоб перший реальний користувач при зверненні до сторінки не зіткнувся з невеликою затримкою, викликаною трансляцією JSP сторінки сервлет і його наступною компіляцією і завантаженням.

Вільно розповсюджуваний Web-сервер Tomcat, реалізація від jakarta.apache.org, автоматично підтримує JSP. дозволяє комбінувати HTML Web-сторінки зі шматочками коду Java в одному документі. Java код обрамляється спеціальними тегами, які говорять JSP контейнеру, що він повинен використовувати код для створення єдиного документа або його частини. Вигода JSP полягає в тому, що можна створити єдиний документ, який представляє і сторінку, і Java код, який у неї включений. Недолік полягає в тому, що той, хто підтримує працездатність JSP сторінки, повинен мати досвід і HTML, і Java (проте, згодом очікується поява візуальних будівельників JSP сторінки).

При першому завантаженні JSP завантажується JSP контейнером (який зазвичай асоційований або є частиною Web-сервера), далі сервлетний код, який обов'язково виділяється JSP тегами, автоматично генерується, компілюється і завантажується в контейнер сервлетів. Статична частина HTML сторінки проводиться у вигляді посилки статичного об'єкта типу String метод write(). Динамічна частина включається прямо в сервлет.

З цього моменту, поки вихідна JSP сторінка не буде змінена, вона буде вести себе так, ніби це була статична HTML сторінка, асоційована з сервлетом (проте весь HTML код насправді генерується сервлетом). Якщо змінити вихідний код для JSP, він автоматично перекомпілюється та перезавантажиться при наступному запиті цієї сторінки[8].

JSP технологія втілює всі динамічні можливості технології JavaServlet, але забезпечує природніший спосіб створення статичного вмісту. Головні особливості JSP технології:

- Мова розробки JSP сторінок, які є текстовими документами та описують, як обробити запит та побудувати відповідь.
- Структурні компоненти доступу до серверним об'єктам.

Механізми визначення розширень мови JSP.

Технологія JSP містить API, який використовується розробниками Web-контейнерів.

Процес моделювання домену є ідентифікацією абстракцій з реального світу. Мета побудови програми на основі абстракцій з реального світу для повторного використання. При моделюванні моделі предметної області, розробка методології «зсередини» використовується. Це означає, що потрібно почати з ключовими об'єктами в системі, а потім рухатися назовні, вивчаючи те, що інші об'єкти, вони взаємодіють с. Таким чином, при визначенні прецедентів або динамічної частини системи, рух відбувається від зовнішньої сторони до внутрішньої, а також при створенні статичної моделі.

Моделювання предметної галузі є основою статичної частини моделі. Побудова моделі предметної області починається з виявлення абстракцій, що у реальному світі, тобто. концептуальних об'єктів, що зустрічаються у системі. При проектуванні об'єктно-орієнтованого програмного забезпечення потрібно структурувати програму так, щоб у центрі опинилися саме ці об'єкти із простору завдання. Це тому, що вимоги до програми змінюються набагато швидше, ніж реальний світ. Основою об'єктного моделювання взагалі і статичного моделювання зокрема є створення моделі цих абстракцій з предметної області. Модель предметної області є словником термінів, яким користуються виявлення та описи прецедентів, об'єктів і класів, які адекватно відображають абстракції предметної області, які у проектованій системі[7].

У результаті виявлення об'єктів з предметної області необхідно встановити, які існують з-поміж них. Дуже важливими зв'язками є відношення агрегації (ставлення між цілим та частиною) та узагальнення (ставлення між підкласом та суперкласом).

На рисунку 2.1. зображена структурна схема веб порталу.

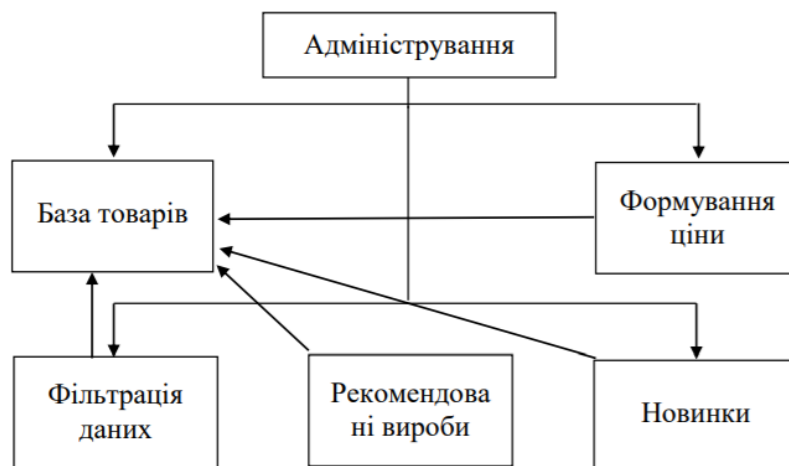


Рис. 2.1. Структура модулів сайту

Словник термінів для створюваної системи:

Звіти – клас алгоритму генерування звітів. Використовується для створення звіту за вибраними даними.

Оператор – конкретний користувач системи. Оператор використовує цю систему для обробки та внесення даних.

БД підприємства - база даних, що містить таблиці з даними інших відділів підприємства, таких як Склад продукції, Транспортні засоби, Клієнти, необхідні безпосередньо для розрахунку планкарти.

До створення системи необхідно вивчити предметну область, тобто, необхідно мати чітке уявлення у тому, як працюватиме система. Діаграма, призначена для створення списку операцій, які виконує система, визначення списку об'єктів системи, які виконують ці функції. Таким чином, будується набір функцій, що виконуються системою, та описуються сценарії поведінки об'єктів, які взаємодіють із системою.

Моделювання прецедентів – процес, спрямований на докладне уявлення дій користувачів та реакцію системи, оскільки її поведінка обумовлена їх вимогами. Тобто. робота системи залежить від цього, як до неї звертаються і чого хочуть домогтися. Вся динамічна частина об'єктної моделі безпосередньо з моделлю прецедентів[5].

Підхід до побудови динамічної моделі називається "зовні всередину". Починається з обстеження користувачів, які знаходяться поза системою, і на шляху розкриваємо деталі поведінки програми. На цій основі створюється структура програми, яка підтримує потрібну поведінку. Просування спрямоване всередину, його кроком є розгляд одного конкретного сценарію.

Діаграма варіантів використання (Use Case Diagram) для електронного магазину, що розробляється, зображена на рис.2.2.

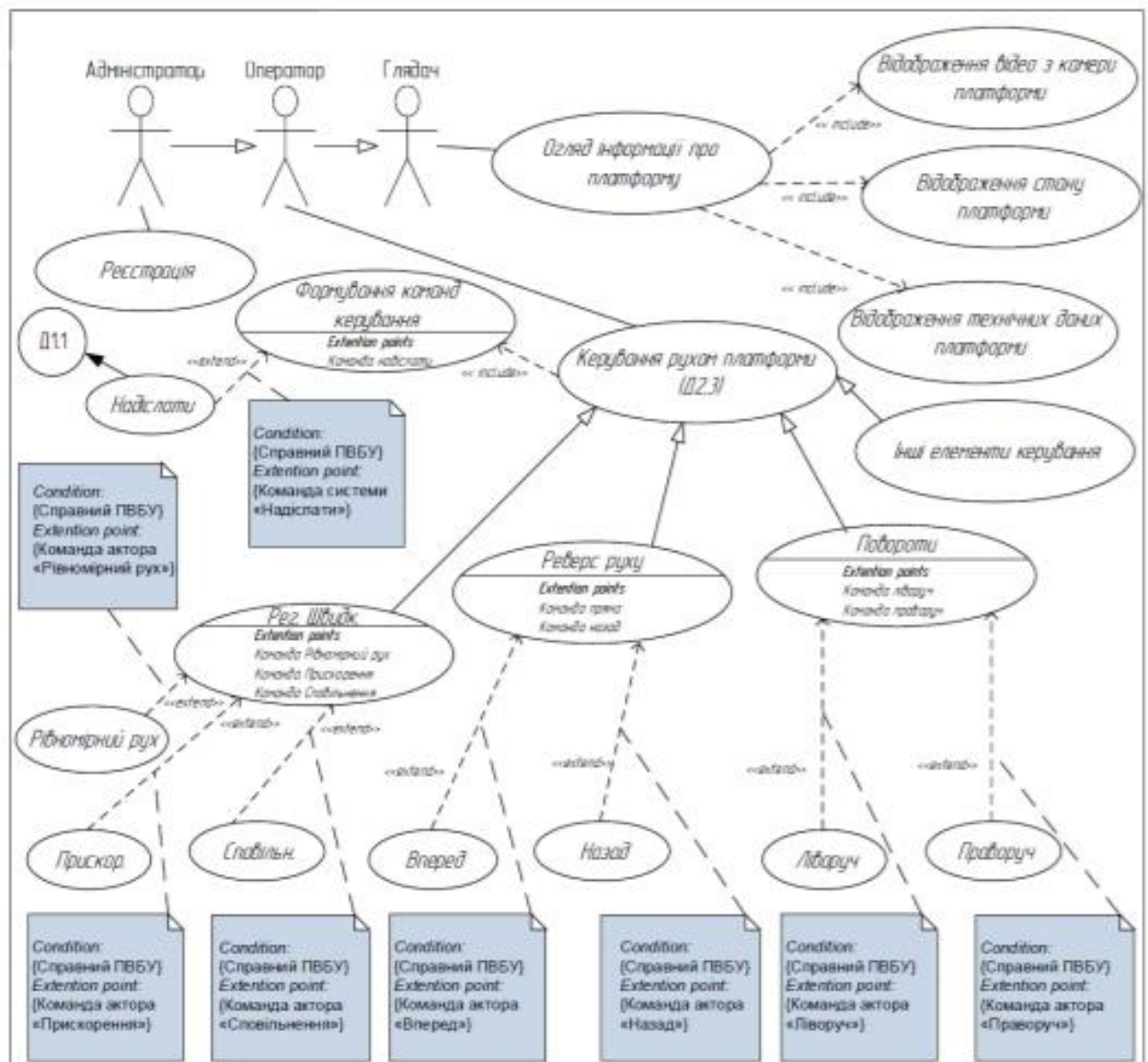


Рис. 2.2. Use Case Diagram

Діаграма послідовності складається з чотирьох основних елементів:

- тексту послідовності дій у прецеденті, який записується зверху вниз ліворуч, цим служить описом дій, що відбуваються під час роботи програми;
- об'єктів, у яких у форматі «об'єкт: клас» записується ім'я або номер екземпляра об'єкта та ім'я класу об'єкта.
- повідомлень, зображуваних стрілками, які спрямовані від одного об'єкта до іншого, що містять супровідну інформацію про дію, що відбувається в певний момент часу і реакції системи на цю дію;
- методів (операцій), які у вигляді прямокутників. Вони розташовані на пунктирних лініях, які відповідають об'єктам, яким належать методи. Довжину прямокутника можна використовувати для того, щоб показати фокус управління в послідовності: метод має управління аж до точки, в якій прямокутник закінчується. Ці прямокутники називаються лінією життя об'єкта. Послідовність дій та відповідна реакція системи на ці дії описані в текстах повідомлень, що передаються між об'єктами[12].

Діаграми потоків даних (Data Flow Diagrams – DFD) являють собою ієрархію функціональних процесів, пов'язаних потоками даних. Мета такого подання – продемонструвати, як кожен процес перетворить свої вхідні дані у вихідні, а також виявити відносини між цими процесами.

Функціональна модель являє собою набір діаграм потоків даних, які описують зміст операцій і обмежень. Діаграма потоків даних відображає функціональні залежності значень, що обчислюються в системі, включаючи вхідні значення, вихідні значення і внутрішні сховища даних. Діаграма потоків даних – це граф, на якому показано рух значень даних від їх джерел через перетворюючі їх процеси до їх споживачів в інших об'єктах (рис.2.3).

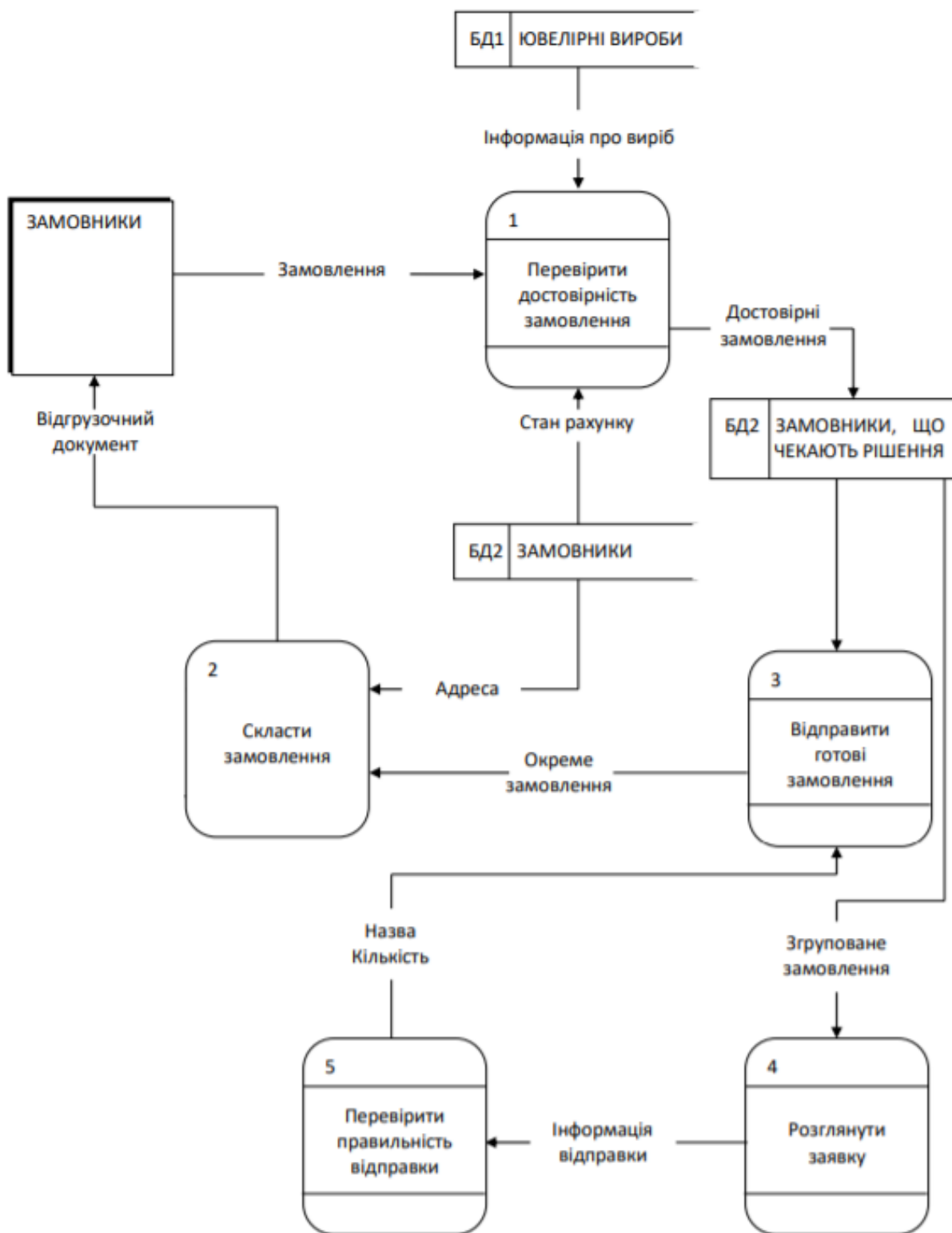


Рис. 2.3. Діаграма потоків даних

Сховище даних та діаграма потоків даних показані на рисунку 2.4 та 2.5 відповідно.

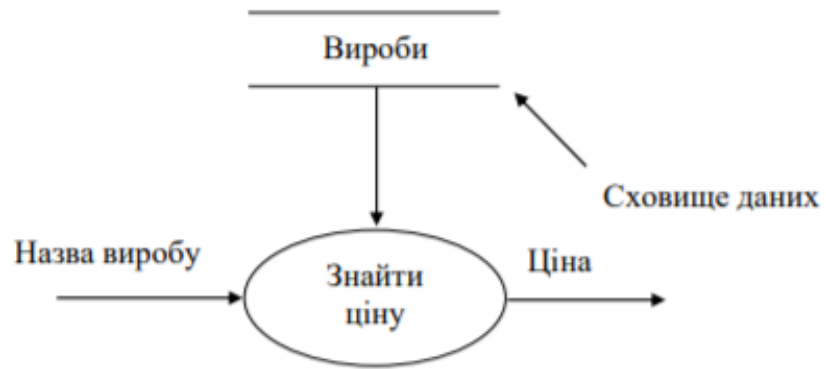


Рис. 2.4. Сховище даних

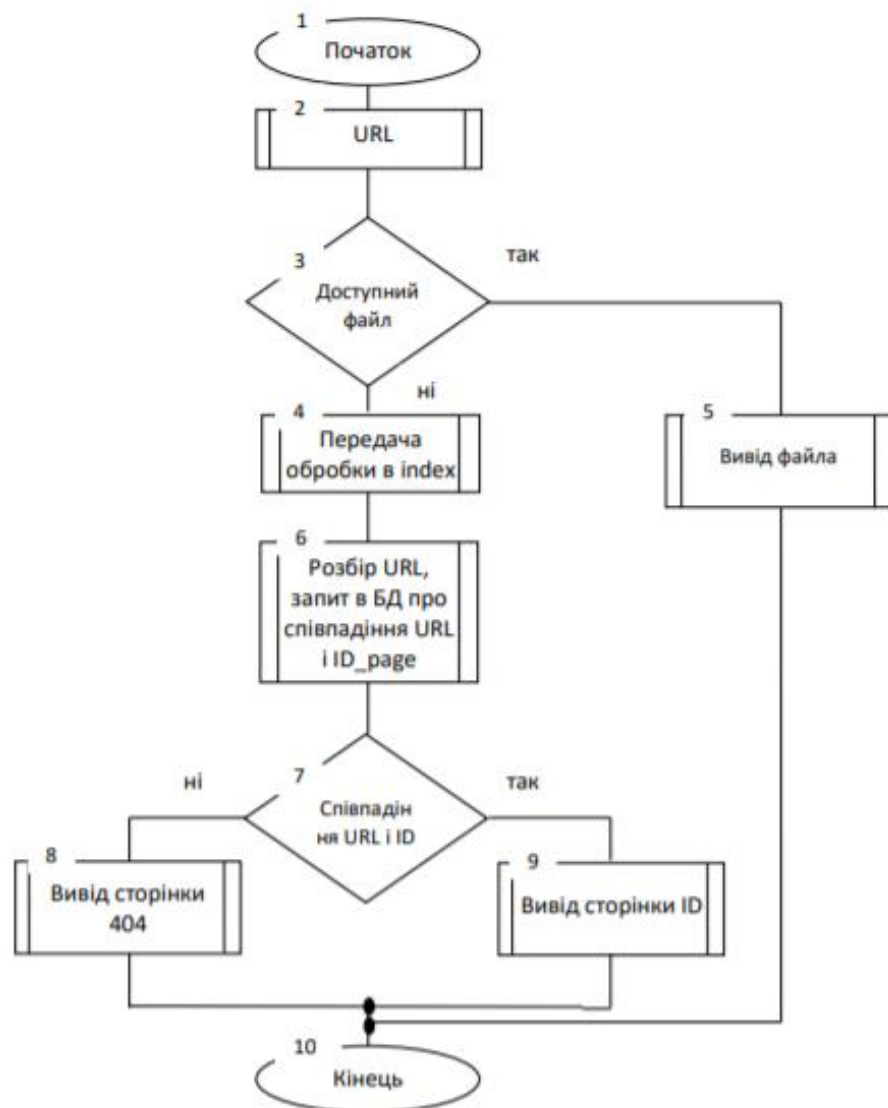


Рис. 2.5. Алгоритм роботи сайту

Діаграма класів – основна діаграма для створення коду програми. З допомогою діаграми класів створюється внутрішня структура системи, описується успадкування і становище класів друг щодо друга. Тут описується логічне уявлення системи, оскільки класи - це заготівлі, основі яких потім будуть визначені фізичні об'єкти і безпосередньо код програми.

Отже, діаграма класів описує загальне уявлення системи, де представлені об'єкти системи. За допомогою діаграми класів можлива зміна в будь-який момент властивостей будь-якого класу або його зв'язків, і при цьому діаграми та специфікації, пов'язані зі змінним класом, будуть автоматично оновлені[27].

Діаграма класів можна використовувати як із аналізі готової системи, і під час розробки нової. Для виявлення класів використовуються різні методи та підходи. Так Барамі (Bahrami) докладно вивчив основні особливості чотирьох основних підходів до виявлення класів (class discovery). Існують такі підходи:

- а) підхід на основі використання іменних груп (тобто іменники в реченнях);
- б) підхід з урахуванням використання загальних шаблонів для класів;
- в) підхід з урахуванням використання прецедентів;
- г) підхід CRC (class – repository – collaborators – клас – обов'язки – «співробітники» – складання специфікації).

У системі, що розробляється, використовується підхід на основі використання прецедентів. Цей підхід рекомендується використовувати в рамках методології RUP (Rational Unified Process). Особливістю такого підходу є проектування знизу – вгору. Після того, як прецеденти відомі, а уявлення про систему частково визначено за допомогою діаграм послідовностей, об'єкти, що використовуються в цих діаграмах, призводять до виявлення класів. Моделювання класів призводить до функціонального підходу, прихильники об'єктно - орієнтованого підходу вважають його проблемно - орієнтованим.

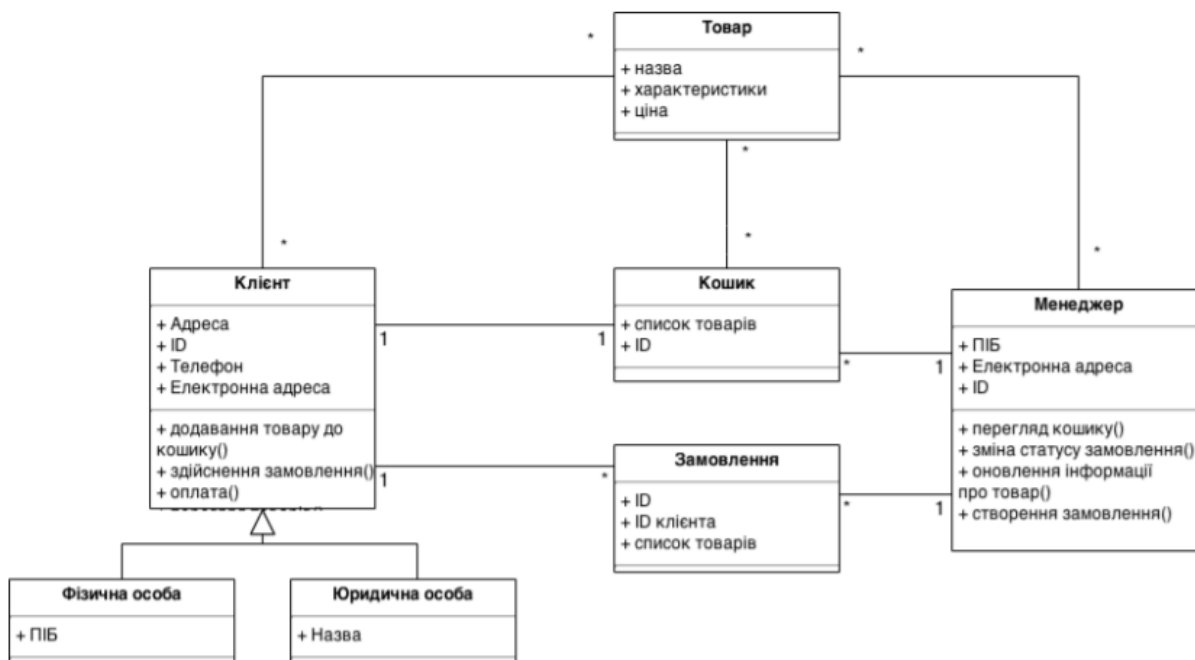


Рис. 2.5. Діаграма класів

Логічне проектування засноване на моделі логічного рівня і являє собою опис та побудову схем зв'язків між елементами даних безвідносно до їхнього змісту та середовища зберігання.

Логічна структура БД виходить перетворенням концептуальної схеми на логічну схему (модель), орієнтовану на обрану СУБД.

Стосовно найбільш поширеної реляційної моделі даних загальний підхід перетворення концептуальної схеми в логічну полягає в тому, що кожному сутності, що є представником безлічі однотипних об'єктів, задають схемою окремого відношення (таблиці), а атрибути сутності утворюють стовпці таблиці. Первинний ключ сутності утворює первинний ключ таблиці, який надалі може бути змінений.

Проектування логічної структури БД має вирішувати завдання вибору найефективнішої структури даних, забезпечення швидкого доступу до даних; виключення дублювання даних, забезпечення цілісності даних таким чином, щоб за зміни одних об'єктів автоматично відбувалася відповідна зміна пов'язаних з ними об'єктів[31].

За неправильно спроектованої схеми БД можуть виникнути аномалії модифікації даних. Вони обумовлені відсутністю засобів явного уявлення типів

множинних зв'язків між об'єктами ПЗ та нерозвиненістю засобів опису обмежень цілісності на рівні моделі даних. Аби вирішити подібних проблем проводиться нормалізація відносин, запропонована Э.Ф. Коддом у рамках реляційної моделі даних (рис.2.6).

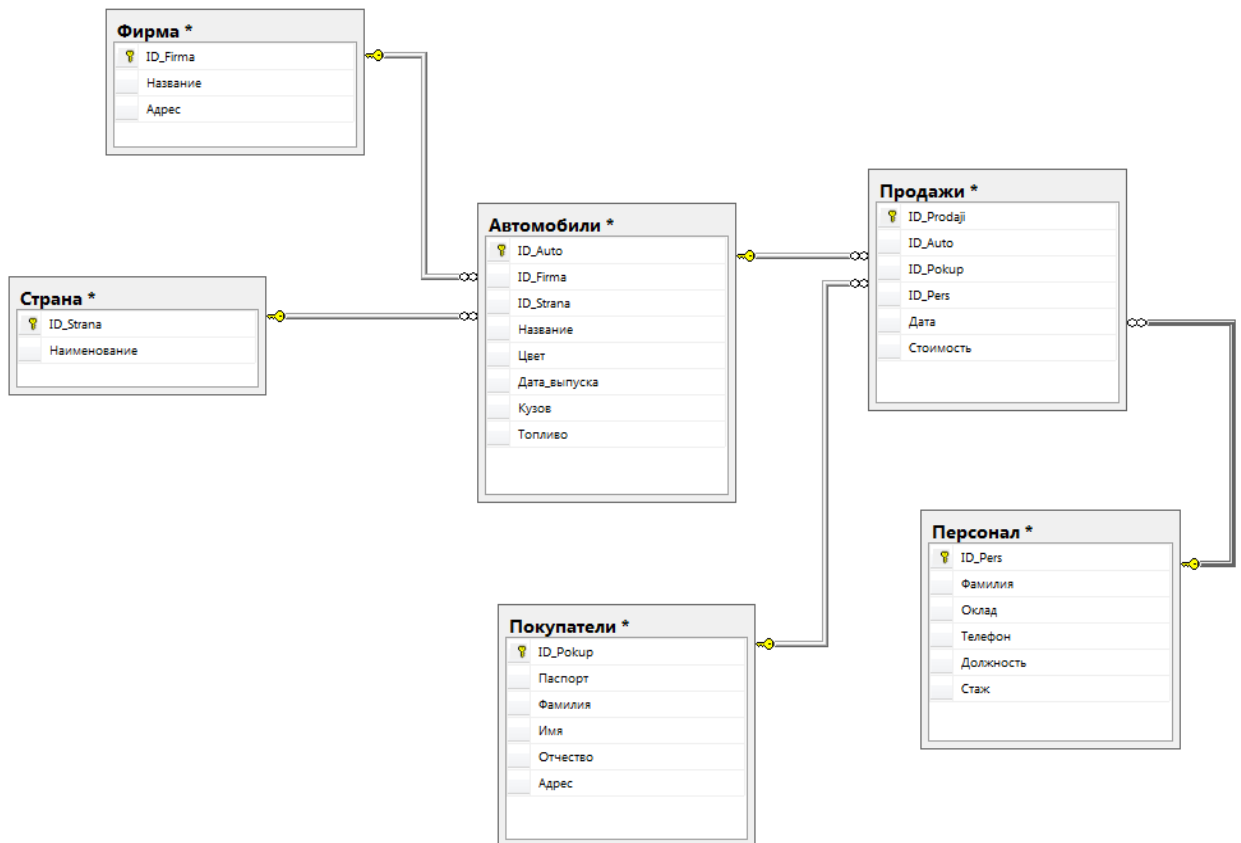


Рис. 2.6. Структура таблиць бази даних в третій нормальній формі

2.4. Реалізація програмного забезпечення

Після завершення етапів аналізу та проектування можна приступати до наступної фази проекту - фізичної реалізації безпосередньо обраною мовою програмування.

Протягом цього етапу необхідно: провести аналіз програмно-апаратної бази користувача та реалізувати результати проектування мовою програмування. Для цих цілей необхідно провести аналіз програмно-апаратної

платформи та середовища розробки для серверної частини, а також мінімальну конфігурацію програмно-апаратної платформи користувача.

Сьогоднішній ринок програмного забезпечення висуває великі вимоги до створюваних проєктів. Так, для сучасних програмних засобів важливими вимогами є переносимість, мультиплатформенність та масштабованість.

Під переносимістю мається на увазі можливість використовувати програмний засіб різних програмно-апаратних платформах без істотної переробки коду.

Масштабованість означає можливість додавання нових функцій та властивостей програмного засобу з мінімальною зміною всього коду загалом. Ідеальним є варіант, який дозволяє нарощувати потужність ПП без зміни основного коду лише додаючи нові модулі.

Тому при розробці враховувалися обидві ці вимоги. Звичайно, створити досить складне ПЗ, яке працювало б на всіх відомих платформах, практично неможливо, але слід прагнути забезпечити його функціональність на найпоширеніших платформах.

Виходячи з того, що платформа IBM PC є найбільш поширеною в Росії, було вирішено розробляти web-сайт саме під цю апаратну платформу. Проаналізувавши системне програмне забезпечення IBM PC-сумісної комп'ютерної техніки, було отримано такі результати: 73% - OS сімейства Windows, 16% - Linux, 11% - BSD, Open BSD, SCO, Mac OS X, Novell NetWare. Виходячи з цих результатів, а також з міркування, що програмне забезпечення має функціонувати на якомога більшій кількості платформ, було прийнято рішення розробляти ПЗ з таким розрахунком, щоб забезпечити функціонування як його окремих компонентів, так і всього комплексу в цілому на двох основних програмних платформах: Windows та Linux. Причому ця вимога поширюється на ту частину програмного забезпечення, що розробляється, яке має бути розміщене на web-сервері[33].

В цей час ринок web-серверів поділений між Apache та Microsoft Information Server (IIS). На їхню частку припадає 85% ринку, найближчі

конкуренти Google та lighttpd мають близько 6%. Цікаво розподіл сайтів, що функціонують під управлінням того чи іншого web-сервера за даними Netcraft [11] - найавторитетнішого джерела в галузі дослідження ринку web-серверів.

Таблиця 2.2.

Статистика використання web-серверів різних виробників

| Розробник | Кількість серверів, жовтень | Доля ринку, % | Кількість серверів, листопад | Доля ринку |
|-----------|-----------------------------|---------------|------------------------------|------------|
| Apache | 68,155,320 | 47.73% | 76,028,287 | 50.76% |
| Microsoft | 53,017,735 | 37.13% | 53,679,916 | 35.84% |
| Google | 7,763,516 | 5.44% | 7,910,879 | 5.28% |
| lighttpd | 1,541,779 | 1.08% | 1,505,122 | 1.00% |
| Sun | 2,262,019 | 1.58% | 619,262 | 0.41% |

У таблиці 2.2. представлена статистика використання web-серверів різних виробників. Як легко можна спостерігати за лідерами належить web-серверу Apache. Які ж його переваги. Розглянемо їх докладніше та порівняємо його з найближчим конкурентом – Microsoft IIS.

Ступінь захисту

Легке досягнення високої безпеки web-сервера Apache у порівнянні з IIS є одним із основних аргументів на користь вибору Apache.

Безпека web-сервера досить багатогранне поняття, яке включає наступні аспекти:

1. Захист від виведення сервера з ладу
2. Захист від розкрадання інформації
3. Захист бази даних у SQL Server

Захист від виведення сервера з експлуатації. Сервер може бути виведений з ладу внаслідок атаки хакерів або дії вірусу. У 2001 році як і Apache, так і IIS виявилися не готовими до нових видів хакерської атаки Denial of service (DoS). Потім віруси класу Code Red вражали сервери Apache та IIS. Слід зазначити, що IIS більш уразливий для хакерів та вірусів, ніж Apache.

Це пов'язано з тим, що IIS не просто web-сервер, а готове бізнес-рішення, яке містить близько 20 web-сервісів різного плану. Іноді вірусу і хакеру вдається вразити одну з IIS (наприклад, Index Search), що призводить до загальної зупинки сервера. В Apache для створення бізнес-рішення доводиться використовувати продукти інших фірм, поразки їхньої системи безпеки не захищаються Apache як прокол. Слід зазначити, що проблеми безпеки слід, не перебільшувати, не применшувати.

Велика кількість служб IIS є проблемою для адміністратора, т.к. якщо всі вони використовуються, потрібно 2-3 десь у місяць встановлювати антивірусні оновлення. В іншому проблему можна вирішити та кардинально, вимкнувши не потрібні IIS служби зовсім. Для цього можна скористатися утилітою IIS Lockdown. Крім того, можна використовувати утиліту AutoUpdate для автоматичної установки оновлень для IIS 6.0 у платформі Microsoft.Net.

Слід зазначити, що більшість поразок серверів Apache та IIS відбувається не так завдяки їхній архітектурі, як з вини адміністраторів, які не працюють над підтримкою безпеки у фірмі. Вбудований захист Apache та IIS не замінить антивірусні програми для web-серверів від Symantec та TrendMicro, та таких

засобів безпеки як Internet Security Scanner. Однак підтримка стійкості web-сервера Apache обходиться адміністратору набагато меншою кількістю дій.

Захист від розкрадання інформації. Якщо у плані стійкості до атак Apache має перевагу перед IIS, то щодо захисту від викрадення інформації IIS багато в чому захищений краще. IIS може використовувати систему ідентифікації користувачів з Windows 2000 Server, це не тільки полегшує адміністрування, а й підвищує захист. На Apache у разі досить типової помилки адміністратора, хакер зможе легко вкрасти список користувачів із паролями.

Слід зазначити, що IIS може використовувати різні засоби шифрування від шифрування диска засобами NTFS до шифрування передачі через SSL. Ці засоби краще реалізовані в IIS, ніж в Apache.

Захист бази даних SQL Server. Оскільки веб-сервер має доступ до корпоративної бази даних у SQL Server, важливим моментом є забезпечення безпеки цієї зв'язки. Microsoft SQL Server 2000 має XML-розширення, які дозволяють передавати дані в Internet через firewall, що дозволяє значно підвищити безпеку.

Розробка

Для Apache стандартні засоби розробки зазвичай представлені засобом створення динамічних сторінок PHP. Можливості продуктів Microsoft тут непорівнянні. У складі IIS 5.0 йде близько 20 різних сервісів на всі випадки життя. Середовище візуальної розробки Microsoft InterDev дозволяє швидко та дешево створювати динамічні web-сервери. Відмінні засоби пошуку помилок (налагодження) Microsoft заслужили широке визнання. Зокрема, IBM рекомендує користуватися інструментарієм Microsoft для налагодження JavaScript у своїх web-серверах. IIS чудово інтегрований з іншими продуктами Microsoft, не буде проблем з організацією взаємодії.

XML-засоби MS SQL 2000 дозволяють значну частину роботи зі створення та підтримки сайту виконати програмістам, що володіють MS SQL.

Продуктивність

IIS 5.0 під Windows 2000 працює значно швидше за Apache під усіма платформами. Також слід зазначити, що за допомогою серверних ActiveX-компонентів можна реалізувати дуже швидкодіючі та складні системи. Наприклад, генерацію графічного зображення з бази даних MS SQL на HTML-сторінку.

У разі використання XML-розширень MS SQL 2000 можна збільшити швидкість IIS в 5-10 разів, т.к. у разі HTML-сторінки генеруються не є інтерпретованим скриптом, а скомпільованим кодом спеціальної бібліотеки.

Однак більшість переваг IIS у швидкодії та багатстві можливостей реалізуються шляхом не зовсім стандартних способів, які підтримую повністю лише браузері на базі MSIE.

Вартість Apache та IIS

Порівнюючи вартість володіння IIS та Apache можна зробити такі висновки:

1. IIS та Apache безкоштовні, якщо ви користувач Windows 2000 Server
2. В IIS значно швидше і дешевше можна розробити або переробити сайт. Розробка сайту буде ще дешевшою, якщо ви використовуєте MS SQL 2000.
3. У Apache дешевшим коштує забезпечення безпеки. Користувачам IIS для зниження вартості безпеки слід відключити непотрібні сервіси, встановити антивірусне програмне забезпечення та використовувати XML-засоби MS SQL 2000.
4. Адміністрування Apache дорожче, ніж IIS. Apache складніше адмініструвати, але адміністратор IIS вимагає частіше встановлювати антивірусні оновлення. Однак установка оновлень для IIS не така дорога і складна робота, як написання конфігураційних файлів для Apache.
5. Персонал. Розробники обійдуться дешевше, крім того, ви можете використовувати для розробки ваших фахівців у MS SQL 2000

Як основний web-сервер для ПП «Інтернет-магазин «Покупка автомобілів»» було обрано сервер Apache. Важливими критеріями для відбору стали мультиплатформенність та безкоштовне розповсюдження. Крім того,

Apache нині один з найбільш популярних web-серверів у всьому світі, тому дуже висока ймовірність того, що він вже входить в серверну конфігурацію системи, на яку буде проводитись установка ПП «Інтернет-магазин «Купівля автомобілів»».

Втім, при необхідності, як web-сервер можна використовувати будь-який сервер, що підтримує взаємодію з PHP.

Середовище розробки для модулів проекту

Сервер повинен забезпечувати доступ до web-форуму та бути досить продуктивним для забезпечення роботи з кількома користувачами. Клієнт повинен мати зручний та звичний для нього інтерфейс. Причому як клієнт, і сервер повинні забезпечувати між собою надійне і захищене з'єднання.

Найочевиднішими є два підходи. Перший передбачає написання окремих додатків сервера та клієнта якоюсь мовою програмування. Але при цьому підході може виникнути наступна проблема: при серйозній зміні структури серверної частини може знадобитися серйозна переробка клієнта, що не завжди буває зручно. Тому розробки серверного ядра було вирішено використовувати трирівневу технологію відому ще як технологія «тонкого» клієнта. Цей підхід передбачає використовувати як клієнтський додаток web-браузер, що підтримує сучасні конструкції мови HTML, що дозволяє організувати розвинений інтерфейс користувача. Як сервер виступає web-сервер з набором скриптів CGI, що реалізує логіку програми і зв'язок даних, що зберігаються на сервері баз даних з інтерфейсом користувача. Це дає такі переваги:

1. Не потрібно встановлювати додаткового клієнтського програмного забезпечення, оскільки web-браузер є практично у будь-якій програмній конфігурації сучасного персонального комп'ютера.

2. Середньому користувачеві не потрібно додаткового навчання для роботи з клієнтською частиною, тому що зазвичай вистачає початкових навичок роботи з web-браузером і навігації по www.

3. Для користувача системи можна вибирати той web-браузер, до якого він звик.

4. Сучасні версії web-браузерів підтримують захищені протоколи обміну даними. Це дозволяє швидко та ефективно вирішити проблему безпечної передачі інформації без додаткового кодування клієнта.

5. Фактично, технологія зводиться лише до створення сервера. При додаванні або зміні функцій сервера не потрібна модифікація коду клієнта, що значно спрощує процес модернізації і найповніше відповідає модульному принципу побудови програми.

Існує кілька видів реалізації серверної частини програми на основі web-сервера: серверний модуль, CGI-додаток, середовища попередньої обробки сторінок, додатки, що вбудовуються. Основне їхнє призначення - організувати інтерактивну взаємодію з користувачем на основі динамічного гіпертексту (DHTML). Вибір конкретного методу обумовлюється різними чинниками, зокрема, розробки проекту нами було висунуто такі вимоги до серверу:

1. Мультиплатформенність та переносимість програмного коду (як і до всього комплексу в цілому) - серверне ядро має без особливих проблем та без зміни вихідного коду функціонувати як на різних системних платформах (Windows, Linux), так і web-серверах.

2. Висока продуктивність.

3. Забезпечення захищеного з'єднання із клієнтом.

Після аналізу засобів розробки ми дійшли висновку, що вищезазначеним вимогам сервер задовольнятиме, якщо як платформи для розробки вибрати PHP.

PHP (PHP: Hypertext Preprocessor) - одна з найпотужніших скриптових мов, доступних на ринку. Він дозволяє легко писати скрипти прямо в HTML файлах і його код буде виконаний транслятором (PHP) на сервері до того, як ця сторінка буде надіслана в браузер. PHP аналогічний концепції Netscape's LiveWire Pro або Microsoft's ASP (Active Server Pages), але за заявою розробників мови, PHP краще обох продуктів. Багато його синтаксису було запозичено з C, Java і Perl з додаванням деяких унікальних особливостей.

Головною метою мови є наданням web-розробникам швидкого створення сторінок, що динамічно генеруються.

Головні можливості:

- PHP безпечний. Використання PHP надає web-серверу ще один рівень між користувачем та сервером, що зменшує можливість користувачам втрутитись у роботу сервера.

PHP має безліч можливостей, які дозволяють програмісту будувати безпечні програми: робота з жорстким кодом у директорії під час компіляції PHP, можливість фільтрувати параметри введення, можливість ігнорувати дані, надіслані GET методом.

- PHP має журнал доступу. При використанні контролю доступу в PHP, користувачі можуть використовувати власні лічильники відвідувань і входу. Ця можливість не використовує центральний системний лог файл доступу і дозволяє отримати моніторинг доступу в реальному часі. Скрипт для перегляду файлу журналу подій дозволяє швидко показувати, наприклад, хто отримував доступ до набору сторінок, що належать конкретному користувачеві або розділу сервера. Пакет може бути налаштований на створення заголовка кожної сторінки з показом інформації про доступ.

Перерахуємо деякі можливості та переваги PHP:

- підтримка великої кількості платформ, включаючи Linux, Microsoft Windows, Mac OS X, FreeBSD, OpenBSD тощо;
- інтеграція з більшістю сучасних web-серверів, наприклад: Apache, Microsoft Internet Information Server;
- робота або як CGI-додатка, або як серверного модуля (для Apache, IIS, Netscape Enterprise Server);
- підтримка як об'єктно-орієнтованого, і процедурного програмування;
- відсутність обмежень на розмір генерованого HTML файлу;
- взаємодія з іншими сервісами та програмами з використанням LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM, CORBA;

- робота із XML документами;
- підтримка найпопулярніших архівних форматів (Zip, BZip2, GZip);
- наявність засобів забезпечення безпечного з'єднання
- повністю безкоштовне OpenSource-розповсюдження.

Дуже важливо і те, що PHP має вбудовану підтримку захисту інформації, що передається по мережі, за допомогою SSL.

Основа розробки — мова програмування JavaScript, що є на сьогоднішній день однією з найпоширеніших мов програмування та найпоширенішою у веб-розробці. Ця мова дає зрозумілий синтаксис, універсальну структуру зберігання даних та можливість писати асинхронний код, який підвищить швидкість роботи програми у рази.

Як платформу було обрано Node.js. Платформа дозволяє використовувати JavaScript не тільки для розробки клієнта у веб-браузері, але й у Backend розробці для написання скриптів сервера. Для спрощення розробки для Node.js існує багато фреймворків.

Використовуваний у цьому додатку Express пропонує баланс між гнучкістю низькорівневої розробки сервера та зручністю використання готових двигунів.

Клієнт був реалізований на патерні SPA (Single Page Application) з фреймворком React.JS, який дозволяє повноцінно використовувати програму, не завантажуючи постійно нові сторінки, а також роботу в офлайн режимі. React.JS — один із найперспективніших сучасних фреймворків, який підтримує багато розробників з усього світу, тому можна буде не турбуватися про розширення функціоналу програми у майбутньому.

Середовищем розробки був обраний JetBrains WebStorm. Вона пропонує одноразову підтримку як React.JS, так і Node.JS Express, швидку роботу та вбудовану підтримку git, хорошу підтримку синтаксису JS та JSX. Інтерфейс програми показаний рисунку 2.7.

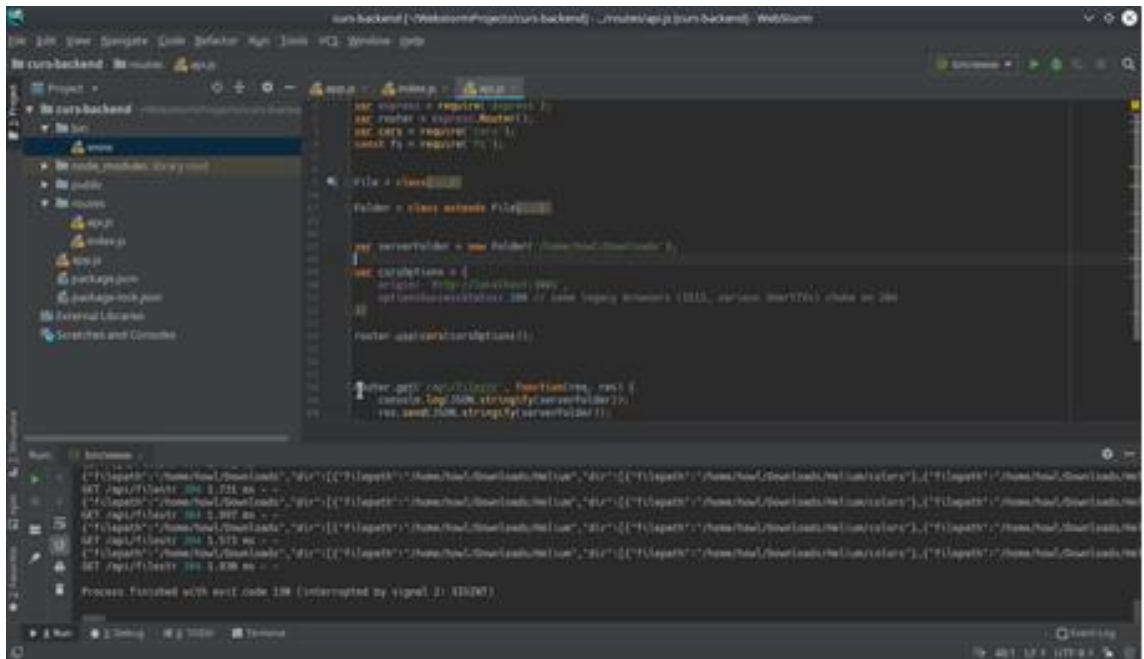


Рис. 2.7. Интерфейс IDE WebStorm.

ВИСНОВКИ ДО РОЗДІЛУ 2

Для розробки веб-сайту застосовувалися:

- мова гіпертекстової розмітки HTML. Застосовується для форматування сторінок веб при відображенні в браузері користувача; формальну мову опису зовнішнього вигляду
- каскадні таблиці стилів CSS. Застосовується для стильового оформлення сторінок веб при відображенні в браузері користувача;
- мова керування сценаріями перегляду гіпертекстових сторінок Web на стороні користувача JavaScript;
- скриптова мова загального призначення, активно використовується для розробки веб-додатків PHP. Ця мова створює динамічні сторінки на вебсайті, власне що важливо, зменшує копіткість роботи.

Як засоби розробки застосовувалися такі методи: Microsoft Visual Studio Code

РОЗДІЛ 3. ДИЗАЙН ІНТЕРФЕЙСУ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1. Структура веб порталу

Сьогодні недостатньо одного розуміння споживачів. Настав час напруженої конкуренції на всіх ринках. Економіка багатьох країн дедалі менше регулюється державою, яка заохочує розвиток ринкових процесів. Європейський союз знищує торгові бар'єри між європейськими країнами, внаслідок чого багато раніше захищених від вторгнення іноземних компаній ринків стають нерегульованими.

Відповідно до концепції маркетингу, компанії досягають конкурентної переваги шляхом розробки пропозицій, які задовольняють потреби цільових споживачів більшою мірою, ніж пропозиції конкурентів.

Компанії можуть надавати більшу споживчу цінність, пропонуючи клієнтам нижчі порівняно з конкурентами ціни на аналогічні товари та послуги або забезпечуючи більше вигод, які виправдовують більш високі ціни. Отже, маркетингові стратегії повинні враховувати як потреби клієнтів, а й стратегії конкурентів. Перший крок у цьому напрямі – аналіз конкурентів. Наступний крок – розробка конкретних стратегій, які дозволяють компанії зайняти міцні позиції у боротьбі з конкурентами та дають найбільшу з усіх можливих перевагу перед конкурентами.

Зі зростанням Інтернету широкого поширення набули веб-інтерфейси, що дозволяють взаємодіяти з різними програмами через браузер (наприклад, управління своїм замовленням в електронному магазині).

| | | | | | | | |
|------------------|-------------|--|--|--|-------------|-------|---------|
| Кафедра КІТ (47) | | | | НАУ 21 26 69 000 ПЗ | | | |
| Розробив | ГончарВ.О. | | | Дизайн інтерфейсу та програмна реалізація | Літера | Аркуш | Аркушів |
| Керівник | МоденовЮ.Б. | | | | | 61 | 21 |
| Консультант | | | | | УС-212М 122 | | |
| Контролер | Райчев І.Е. | | | | | | |
| | | | | | | | |

Веб-інтерфейси зручні тим, що дають змогу вести спільну роботу співробітникам, які не знаходяться в одному офісі (наприклад, веб-інтерфейси часто використовуються для заповнення різних баз даних або публікації матеріалів в інтернет-ЗМІ).

Інтерфейс - у сенсі - певна стандартами межа між взаємодіючими незалежними об'єктами. Інтерфейс задає параметри, процедури та характеристики взаємодії об'єктів.

Інтерфейс користувача - елементи та компоненти програми, які здатні впливати на взаємодію користувача з програмним забезпеченням. В тому числі:

- засоби відображення інформації, інформація, формати та коди, що відображається;
- командні режими, мова користувача-інтерфейс;

Дані вхідні пристрої та технології;

- діалоги, взаємодія та транзакції між користувачем та комп'ютером;
- зворотний зв'язок із користувачем;
- підтримка прийняття рішень у конкретній предметній галузі;
- порядок використання програми та документація на неї.

Є кілька простих правил, що дозволяють інтерфейс магазину бути зрозумілим клієнту.

Що простіше, то краще. Це зовсім не означає, що має бути тільки текст та контактна інформація, але на сайті не має бути марної інформації, шрифт має легко читатись. Графічні елементи мали бути чіткими, виразними і швидко завантажуватися. На сайті магазину неприпустимо використовувати анімацію та звук, які довго завантажуються та відволікають увагу покупця.

Людське око переглядає сторінки сайту зверху донизу. Найбільша увага зосереджується на верхній лівій частині сторінки. Тому у верхній частині сторінки сайту, як правило, розміщується найважливіша інформація: назва фірми, логотип, сама назва сайту і т.д.

Чим простіше верхня частина сторінки, тим легше запам'ятати назву сайту і самої компанії.

Типова помилка багатьох сайтів є нагромодження у верхній частині сторінки сайту з авангардних і складної композиції багатьох малюнків, серед яких назва компанії і назва сайту, які ледь помітні. В результаті людське око, скануючи дану частину сторінки сайту, не встигає розпізнати і запам'ятати необхідну інформацію. Надмірна відбувається насичення. Замість збільшення запам'ятовування, на таких сайтах, є зниження запам'ятовування основних написів (малюнків) сайту, назва компанії і назва сайту.

. Покупець повинен легко знаходити цікавий для нього товар і мати можливість полікувати про нього вичерпну інформацію (опис у вигляді тексту плюс кілька фотографій). На вітрині бажано розмістити дані про діяльність фірми-продавця (що займається компанія, які регіони обслуговує, адресу, контактну інформацію тощо).

Товари мають бути розподілені за групами. Необхідно забезпечити можливість пошуку товарів щодо назви та опису. Для кожного товару зазвичай передбачені: короткий та повний опис, плюс кілька фотографій.

При перегляді товарних пропозицій, покупець повинен мати можливість сортувати товар за ціною або за назвою.

Для наочності необхідно передбачити спеціальні розділи, що містять товари, що згруповані за маркетинговими ознаками. Допустимо:

- «Новинки» (товари, що недавно надійшли у продаж);
- «Спеціальні пропозиції» (товари, на які з якихось причин знижено ціни);
- «Товари дня» (наймодніші товари);
- «Лідери продажів» (найбільш куповані товари).

При оформленні замовлення покупець повинен запровадити контактну інформацію: логін, пароль, адресу доставки, телефон тощо.

Розрахунок вартості та виведення цін має здійснюватися у тензі. Щоб заощадити час замовника на оплату товару, непогано мати в магазині готову квитанцію для оплати через банк.

. В електронному магазині можуть бути й інформаційні розділи:

з даними про компанію (сфера діяльності, адреса, контактні телефони тощо);

з інформацією щодо доставки товару;

з інформацією щодо знижок;

новини електронного магазину;

статті (системи управління новинами та статтями надають можливість використовувати електронний магазин як справжній інформаційний портал);

інша корисна інформація.

Розсилка новин. Відвідувач має можливість підписатись (і відписатись) на новини інтернет-магазину. Після підписки покупцю періодично надсилається інформація про новинки магазину.

Розділ «Питання та відповіді» (FAQ), де покупець може поставити своє запитання, а адміністратор магазину – відповісти на нього. Це сприяє підвищенню якості обслуговування, утриманню старих клієнтів та залученню нових.

«Відгуки та питання» до товарів. Покупцеві слід надати можливість переглянути відгуки про товар, залишити свій відгук або поставити питання - це сприяє підвищенню якості опису товарів.

Збільшенню відвідуваності електронного магазину та кількості постійних клієнтів сприяє також наявність зворотного зв'язку адміністратора з клієнтами. Важливо, щоб адміністратори якнайшвидше та максимально повно відповідали на листи.

Зворотною, невидимою покупцю стороною електронного магазину є система управління. Вхід у систему адміністрування здійснюється лише після введення адміністратором логіну та паролю (логін та пароль адміністратор

може змінювати). Адміністратор має можливість повністю керувати вмістом електронного магазину:

- додати або видалити продукти, описи та фотографії до них, змінювати їх вартість, терміни поставки товарів і рівень знижок;
- редагувати розділи магазину (новини, статті, запитання та відповіді, огляди та питання про товари тощо);
- редагувати спеціальні розділи магазину (новинки, спеціальні пропозиції, товари дня, лідери продажу);
- редагувати контактну інформацію електронного магазину;
- редагувати зміст заголовків та текстів листів, що відправляються покупцеві при реєстрації та купівлі товару;
- складати та розсилати листи з новинами магазину передплатникам;
- переглядати історію замовлень і клієнтів статистики;

Якщо компанія має намір працювати ще й з оптовими клієнтами, необхідно передбачити роботу сайту як із роздрібними, так і з оптовими цінами на товари.

. Завантаження товарних пропозицій має здійснюватись із файлу Microsoft Excel. Файл, в якому описані всі групи, підгрупи каталогу, а також інформація про товари, створюється на локальному комп'ютері (обмежень кількості груп, категорій і самих товарів бути не повинно). Потім файл завантажується до інтернет-магазину. Після завантаження прайс-листа, товари автоматично з'являються на вітрині. Таким чином, можна оперативно змінювати дані про ціни та наявність товарів.

. Акуратна робота із кольором. Правильно застосований колір може, наприклад, передавати тонкі різниці між однорідними елементами. Неправильно використаний колір може заважати працювати з програмою.

Особливо це стосується червоного кольору. Так вийшло, що для всіх людей червоний колір асоціюється з якоюсь небезпекою. Велика кількість

червоного кольору в якомусь місці на екрані привертає увагу, змушує користувача насторожуватися, думаючи, що щось не так.

Дорожні знаки червоного кольору або забороняють або попереджають про небезпеку. Тому, якщо кнопка на екрані пофарбована червоним, незалежно від того, що на ній написано, користувач намагатиметься уникати натискання на неї.

У малих кількостях червоний колір може справно служити ненав'язливою вказівкою наявності будь-яких проблем. Наприклад, якщо число, що вийшло в результаті розрахунку, перевищує норму.

Червоний колір також можна використовувати в парі з іншими. Є два метафор - «термометр», коли синій відміну від червоного, і «світлофор» - зелений. В обох випадках слід використовувати тільки в разі потреби.

Керуючись даними принципами розробки інтерфейсу, було вирішено зробити ставку на простоту та інформативність, щоб користувач, потрапляючи на сайт, повинен отримувати чітку інформацію про товар, новинки, майбутні релізи. Також про те, як він зможе оплатити замовлення, які умови та терміни доставки тощо.

В електронному магазині має бути реалізований зручний та швидкий пошук необхідного користувачеві товару, тому що не всі мають необмежений доступ до Інтернету, і багато хто оплачує його по годинах. Та й стомлюючий перегляд каталогів мало кому до душі.

Всі товари, які пропонуються на сайті, повинні бути в наявності. Оскільки навряд чи когось порадує повідомлення про те, що замовлений товар, він зможе отримати лише за місяць. Буде передбачено систему замовлення товару, відсутнього на складі.

Буде зроблена проста і гранично зручною навігація, щоб користувач оперативно отримував відповіді, на будь-які питання по тематиці сайту. Не знайшовши потрібної інформації, відвідувач може просто піти, так і не отримавши чіткого уявлення про ваш ресурс.

Сторінки сайту були реалізовані за допомогою HTML-тегів, динамічне відображення форм. Також розробки сторінок використовувалися таблиці, посилання як зовнішні, і внутрішні.

Web-форум був реалізований за допомогою PHP, містить у собі дві сторінки: перша використовується користувачем, друга використовується адміністратором. Усі дані, що знаходяться у форумі, доступні для перегляду та редагування на сторінці адміністрування.

Зовнішній вигляд веб порталу зображено на рисунку 3.1.

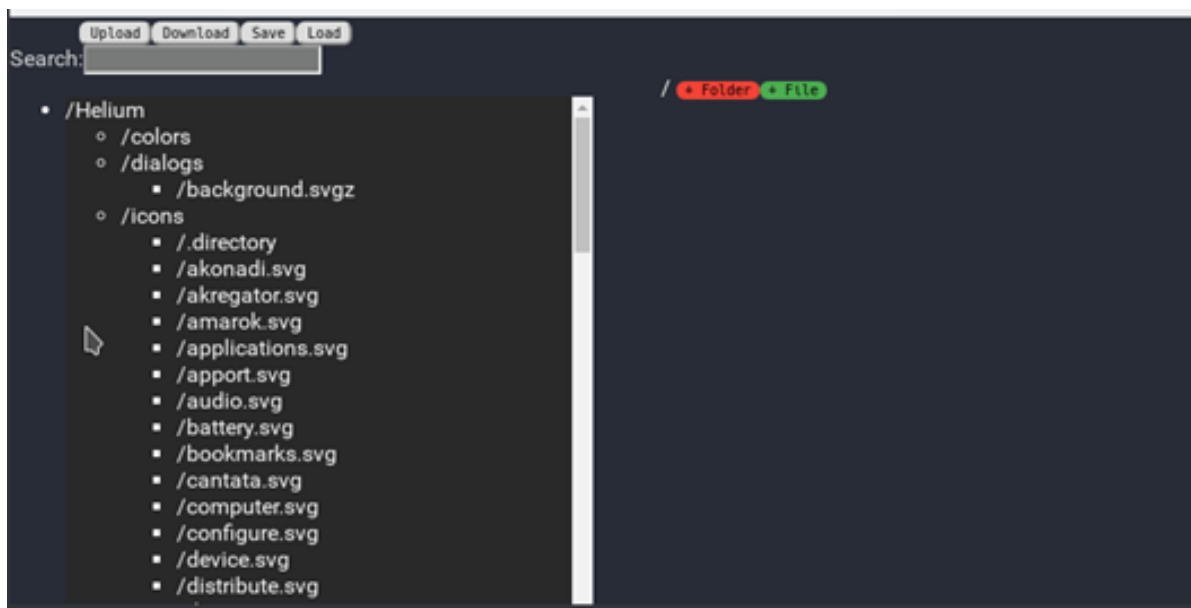


Рис. 3.1. Зовнішній вигляд веб порталу

Демонстрація тестової структури представлена на рисунку 3.2.

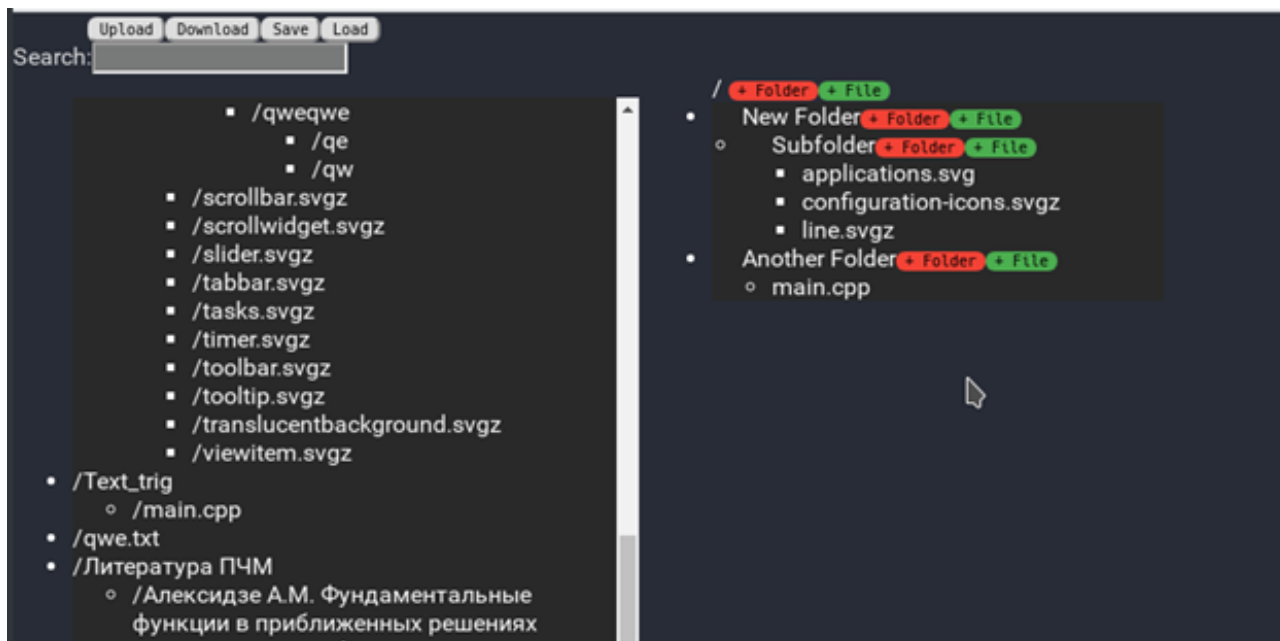


Рис. 3.2. Демонстрація тестової структури

Кнопка (+Folder) у правій частині програми дозволяє додати папку до структури, представленої таблицею. При натисканні вона викликає поле введення назви папки та додаткову кнопку для підтвердження. Це показано на рисунку 3.3.

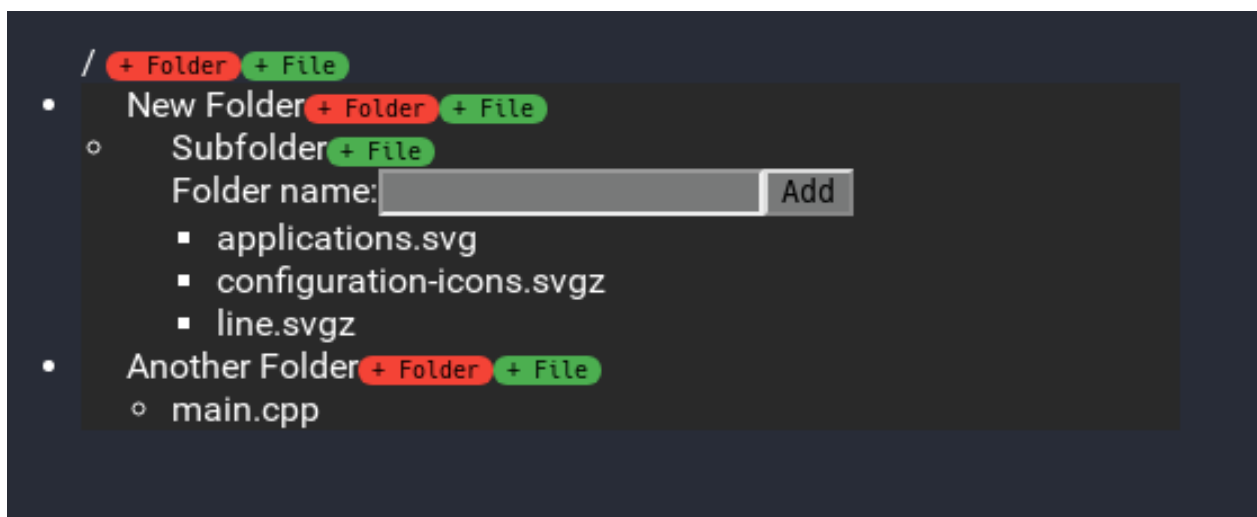


Рис. 3.3. Принцип роботи кнопки Folder

Додавання файлу до необхідної натискання кнопки (+File) . Попередньо треба вибрати файл у лівій таблиці, просто клацнувши по ньому.

Додавання файлів та папок реалізовано за допомогою рекурсивних функцій та відповідають стандартам React, тобто вони не змінюють стан структури безпосередньо, а попередньо створюється копія стану (Додаток А).

Знаходження файлу по кліку в лівій таблиці також було реалізовано за допомогою підхоплення події натискання та виклик рекурсивної функції пошуку в дереві.

Далі завантажимо збережемо структуру на сервері як тестового XML-файла. Натиснемо на кнопку Upload і на сервері отримаємо XML-файл, що з'єднується, представлений на рисунку 3.4.

```
<filepath>/</filepath>
<active>>false</active>
<dir>
  <filepath>New Folder</filepath>
  <active>>false</active>
  <dir>
    <filepath>Subfolder</filepath>
    <active>>false</active>
    <dir>
      <filepath>applications.svg</filepath>
      <realloc>/home/howl/Downloads/Helium/icons/applications.svg</realloc>
    </dir>
    <dir>
      <filepath>configuration-icons.svgz</filepath>
      <realloc>/home/howl/Downloads/Helium/widgets/configuration-icons.svgz</realloc>
    </dir>
    <dir>
      <filepath>line.svgz</filepath>
      <realloc>/home/howl/Downloads/Helium/widgets/lineedit.svgz</realloc>
    </dir>
  </dir>
</dir>
<dir>
  <filepath>Another Folder</filepath>
  <active>>false</active>
  <dir>
    <filepath>main.cpp</filepath>
    <realloc>/home/howl/Downloads/Text_trig/main.cpp</realloc>
  </dir>
</dir>
```

Рис. 3.4. Отримана в результаті XML структура

Розглянемо наочно основні пункти виконання роботи на рисунках 3.5 – 3.8

```

FindAndPaste(where, whatf, whatp){
  if(where.dir){
    if(where.dir.findIndex(this.isRight, whatf) !== -1){
      console.log(where);
      let whatplast = whatp.split("/");
      where.dir[where.dir.findIndex(this.isRight, whatf)].dir.push({filepath: whatplast[whatplast.length-1], realloc: whatp});
      return 0;
    }
    else{
      where.dir.forEach((el) => {
        this.FindAndPaste(el, whatf, whatp);
      });
    }
  }
}

```

Рис. 3.5. Процедура вставки файлі в дерево.

```

addFolder(el, id, name){
  if (el.filepath === id){
    el.dir[el.dir.length] = {filepath: name, active: false, dir: []};
    this.forceUpdate();
    return true;
  }
  else {
    if(el.dir)
    el.dir.forEach((elem) => {
      if(elem.filepath === id){
        elem.dir[elem.dir.length] = {filepath: name, active: false, dir: []};
        this.forceUpdate()
      }
      else {
        this.addFolder(elem, id, name);
      }
    })
  }
}

```

Рис. 3.6. Додавання папки в дереві.

| | | | |
|--------------------|------------------|--------------------|--------|
| backend | 30.01.2020 15:53 | Папка с файлами | |
| common | 30.01.2020 15:53 | Папка с файлами | |
| console | 30.01.2020 15:53 | Папка с файлами | |
| environments | 30.01.2020 15:53 | Папка с файлами | |
| frontend | 30.01.2020 15:53 | Папка с файлами | |
| vagrant | 30.01.2020 15:53 | Папка с файлами | |
| vendor | 30.01.2020 16:02 | Папка с файлами | |
| .bowerrc | 30.01.2020 15:53 | Файл "BOWERRC" | 1 КБ |
| .gitignore | 30.01.2020 15:53 | Текстовый докум... | 1 КБ |
| codeception.yml | 30.01.2020 15:53 | Файл "YML" | 1 КБ |
| composer.json | 30.01.2020 15:59 | Файл "JSON" | 2 КБ |
| composer.lock | 30.01.2020 16:02 | Файл "LOCK" | 138 КБ |
| docker-compose.yml | 30.01.2020 15:53 | Файл "YML" | 1 КБ |
| init | 30.01.2020 15:53 | Файл | 9 КБ |
| init.bat | 30.01.2020 15:53 | Пакетный файл ... | 1 КБ |
| LICENSE.md | 30.01.2020 15:53 | Файл "MD" | 2 КБ |
| README.md | 30.01.2020 15:53 | Файл "MD" | 3 КБ |
| requirements.php | 30.01.2020 15:53 | Файл "PHP" | 6 КБ |
| Vagrantfile | 30.01.2020 15:53 | Файл | 3 КБ |
| yii.bat | 30.01.2020 15:53 | Пакетный файл ... | 1 КБ |

Рис. 3.7. Ієрархічна структура файлу

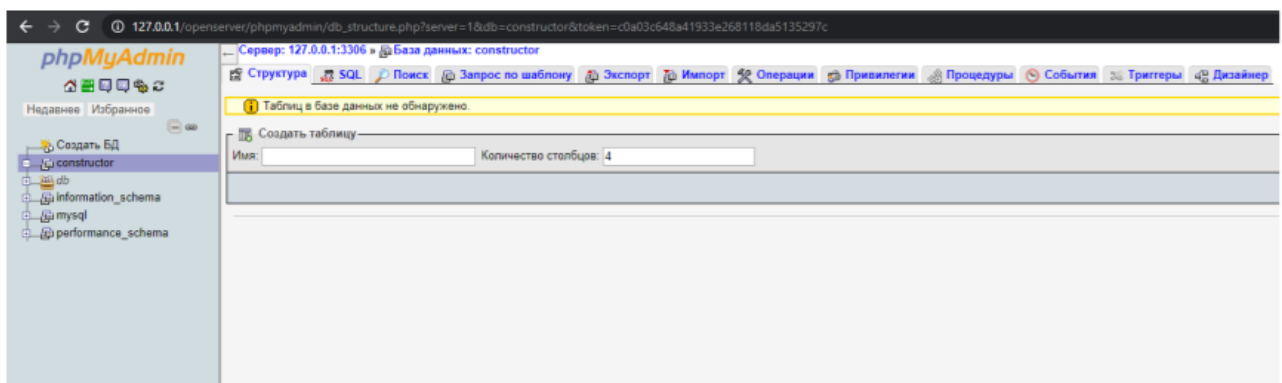


Рис. 3.8. Інтерфейс phpMyAdmin

3.2. Програмна реалізація

При розборі архітектури Інтернет-магазину для зручності було позначено кілька частин: адміністраторська, клієнтська та програмна.

Адміністраторська частина містить інструменти управління Інтернет - магазином і включає як загальні налаштування магазину, так і спеціальні налаштування.

У клієнтській частині архітектури розробляється максимально зручна та доступна робота потенційного клієнта на сторінках Інтернет-магазину. Розробка інтерфейсу, доступні та зрозумілі діалогові вікна, зручні системи оплати та доставки товарів.

Програмна частина архітектури Інтернет-магазину розглядається як взаємозв'язок операційної та серверної частини.

Серверна частина містить у собі розміщення Інтернет-магазину на сайті провайдера, що підтримує технології, що використовуються при створенні Інтернет-магазину.

В операційній частині розглядається середовище розробки Інтернет-магазину.

Для створення Інтернет-магазину надається перевага php. Це потужне середовище для розробки, сумісне з усіма операційними системами та браузерами, що не потребує високих апаратних засобів комп'ютера, досить просте в освоєнні і продовжує розвиватися та вдосконалюватися. Також він підтримується переважною більшістю платних хостингів, що є безперечним плюсом.

Вибір платного хостингу полягає в тому, що є хоч якісь гарантії, сайт отримує ім'я на доменному рівні, підтримуються всі сучасні технології, не буде настирливих рекламних банерів, що не належать до тематики сайту, швидкість закачування буде помітно вищою, обслуговування таких сайтів зручніше, є можливості для розвитку, запровадження нових послуг для залучення клієнтів. Також можна укласти довгостроковий договір, що гарантуватиме безперебійну роботу сайту, його захист від злону та вірусів, дозволить уникнути неприємних сюрпризів на кшталт припинення існування цього хостингу.

На початковому етапі в проєкті Інтернет-магазину реалізуються такі способи оплати, як оплата поштовим переказом, оплата кур'єру готівкою, за кредитною картою, за допомогою Інтернет-банкінгу та інтернет гаманця.

Проте електронна торгівля розвивається не досить активно. Банкіри не виявляють ініціативи, тому що немає привабливого для них обігу. Обороти немає, тому що не розроблені достатньо надійні та дешеві рішення. Рішень немає, бо немає інвестицій у електронну торгівлю. Виходить замкнене коло, яке можна розірвати лише спільно – електронним торговцям (зараз це провайдери Internet), представникам банків та виробникам програмного та апаратного забезпечення.

Відаповідно до описаної у попередніх розділах структури маємо наступний зовнішній вигляд компонентів (рис. 3.9 – рис.3.12).

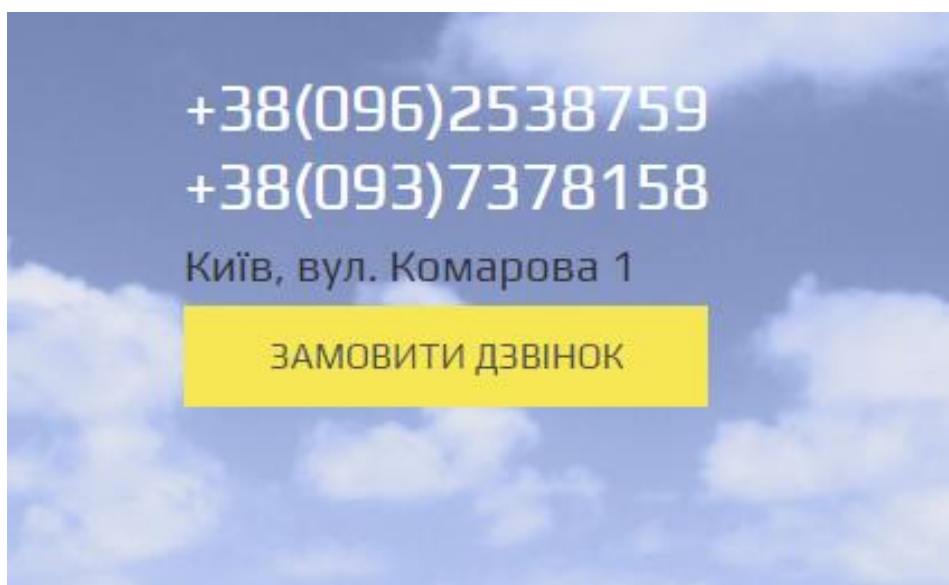


Рис. 3.9. Вигляд вікна «Зв'язатися»



Рис. 3.10. Центральне випадаюче меню

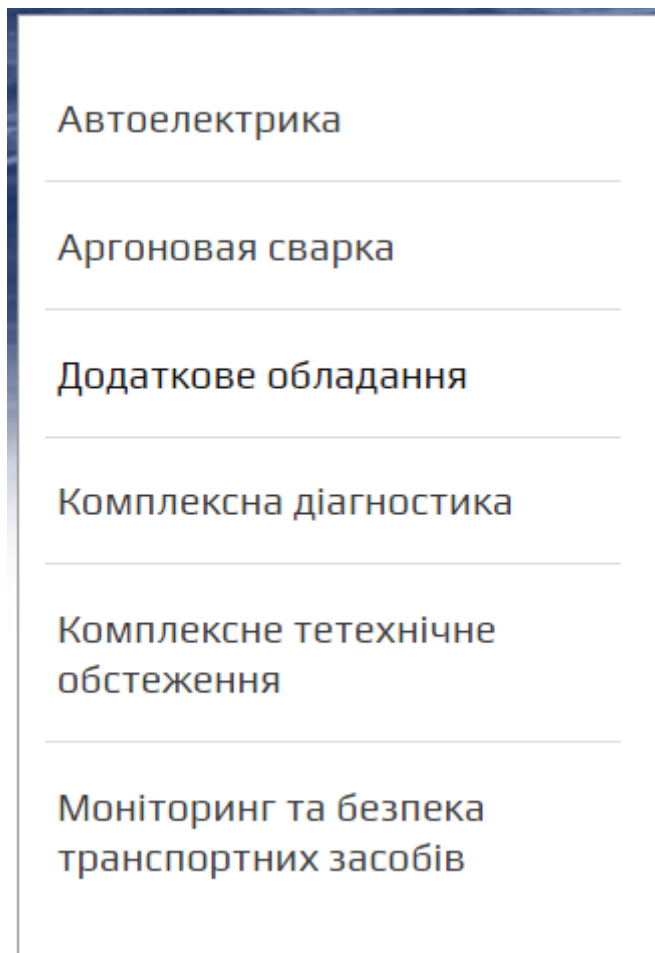


Рис. 3.11. Бокове меню по вибору пропонованих послуг

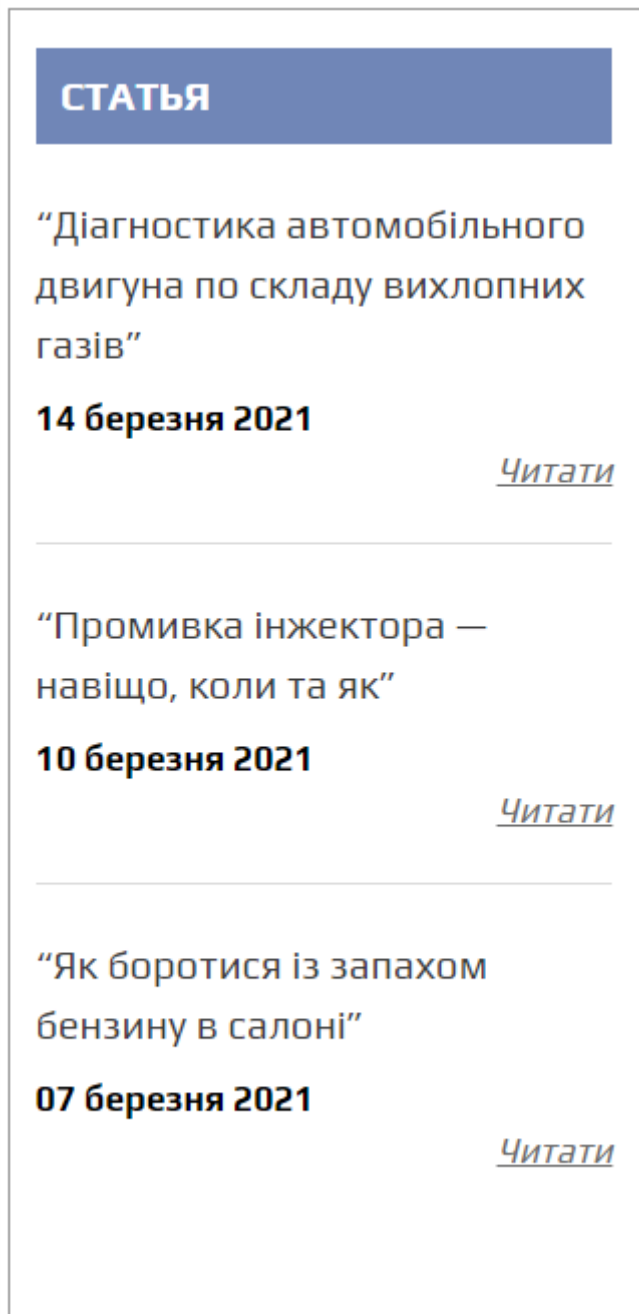


Рис. 3.12. Розділ із корисними та клікабельними статтями

3.3. Оцінка розробленого програмного продукту

Тестування відіграє життєво важливу роль у процесі розробки та створення якісного програмного забезпечення. Необхідно серйозно ставитися

до аналізу та проектування структурованого процесу, який забезпечить своєчасний та успішний випуск проекту.

Важливо пам'ятати, що довіра користувачів дуже просто втратити, і виправити помилки може коштувати дорожче, ніж спочатку зробити повну підготовку і тестування.

Етапи тестування веб-проектів:

Підготовчий етап та вивчення документації.

На даний етап входить аналіз технічного завдання; вивчення кінцевих макетів; тест кейсів; матриці відповідності (для валідації покриття вимог щодо продукту тестами) та складання плану тестування.

Не існує методів тестування сайтів, які б дозволили зі стовідсотковою ймовірністю виявити помилки в роботі програмного забезпечення (продукту). Існує лише формальний процес проведення тестування. Відповідно до цього процесу є можливість довести, що у програмному забезпеченні немає помилок лише з точки зору обраного методу тестування. Таким чином виходить, що немає можливості встановити та гарантувати, що в програмі немає помилок або дефектів, які можуть виникнути через людський фактор, присутній на всіх етапах життєвого циклу програмного забезпечення. На сьогоднішній день існує велика кількість підходів до тестування програмного забезпечення.

Підходи до тестування сайтів:

Звичайна практика полягає в тому, що після закінчення розробки ПЗ і до передачі товару замовнику проводиться тестування сайту. Зазвичай вона виявляється у вигляді окремої фази тестування у загальному циклі розробки ПЗ.

Тестування сайтів розпочинається одночасно зі стартом розробки продукту. Продовжує весь той час, поки триває розробка. Цей підхід потребує більше ресурсів, але якість тестування помітно вища.

Рівні тестування веб-сайтів, які можна виділити:

Модульне тестування - тестується мінімально можливі компоненти веб-сайту. Клас, функція тощо.

Інтеграційне тестування - шукає проблеми в інтерфейсах (не користувальницьких уявленнях даних) взаємодії між модулями системи.

Функціональне тестування – перевіряється відповідність системи вихідним вимогам до неї.

Тестування сайту у різних браузерях.

Варто окремо відзначити, що під час функціонального тестування веб-сайтів обов'язково потрібно перевірити коректність роботи веб-програми, а також однаковість дизайну у всіх сучасних браузерах[12].

Як під час розробки програмного забезпечення, так і після її закінчення необхідно проводити аналіз якості розробленого програмного забезпечення, що надалі дозволить визначити якість та працездатність кінцевого продукту. Аналіз якості сприяє виявленню та виправленню максимальної кількості помилок при створенні ПЗ, що веде до максимальних показників якості.

Якість програмного забезпечення може бути оцінена наступними характеристиками (Набір характеристик оцінки якості ПЗ визначений ГОСТом):

Функціональні можливості. Набір атрибутів, які стосуються суті набору функцій та його конкретним властивостям. Функціями є ті, що реалізують встановлені чи передбачувані потреби:

Програмний продукт забезпечує свої функціональні можливості якщо він має набір функцій для вирішення завдань відповідно до його призначення, враховуючи отримання коректних результатів. Якщо програмний продукт є відкритою системою і має можливість взаємодії з іншими програмними засобами, за умови що він також має здатність запобігати несанкціонованому доступу як випадковий, так і навмисний, до програм та даних.

Практичність. Набір атрибутів, що стосуються обсягу робіт, необхідних для використання та індивідуальної оцінки такого використання певним чи передбачуваним колом користувачів.

При цьому користувачі можуть інтерпретуватись як більшість безпосередніх користувачів інтерактивного програмного забезпечення. Коло користувачів може включати операторів, кінцевих користувачів та непрямих користувачів, на яких впливає програмне забезпечення або які залежать від його використання. Практичність повинна розглядатися у всій різноманітності умов експлуатації користувачем, які можуть впливати на програмне забезпечення, включаючи підготовку до використання та оцінку результатів.

Ефективність. Набір атрибутів, що відносяться до співвідношення між рівнем якості функціонування програмного забезпечення та обсягом ресурсів, що використовуються за встановлених умов.

Мобільність. Набір атрибутів, що належать до можливості програмного забезпечення перенести з одного оточення в інше. Примітка - навколишнє середовище може включати організаційне, технічне або програмне оточення.

У процесі розробки веб-сайту було виконано всі вимоги, які пред'являли замовник.

Тестування розробленого програмного забезпечення

Задля більшої якості програмного продукту тестування застосовувалося кожної стадії розробки.

Якість програмного продукту характеризується набором властивостей, що визначають, наскільки продукт "добрий" з точки зору зацікавлених сторін, таких як замовник продукту, спонсор, кінцевий користувач, розробники та тестувальники продукту, інженери підтримки, співробітники відділів маркетингу, навчання та продажу. Кожен із учасників може мати різне уявлення про продукт і про те, наскільки він хороший чи поганий, тобто про те, наскільки високою є якість продукту. Таким чином, постановка задачі забезпечення якості продукту виливається в завдання визначення зацікавлених осіб, їх критеріїв якості та потім знаходження оптимального рішення, що задовольняє цим критеріям. Тестування є одним із найбільш усталених способів забезпечення якості розробки програмного забезпечення та входить до набору

ефективних засобів сучасної системи забезпечення якості програмного продукту.

З технічної точки зору тестування полягає у виконанні додатка на деякій кількості вихідних даних і звірванні одержуваних результатів із заздалегідь відомими (еталонними) з метою встановити відповідність різних властивостей та характеристик додатка замовленим властивостям.

Тестування на стадії аналізу та дослідження предметної галузі.

На цьому етапі перевірялася повнота вимог до програмного забезпечення: перевірено функціональний склад програмного продукту. У процесі узгодження функціонального набору програмного забезпечення із замовником, список функцій неодноразово змінювався, доки було визначено розробниками разом із замовником, що ці функції враховані і більше жодних виправлень не вимагається.

Тестування на стадії реалізації програмного продукту.

Модульне тестування – це тестування програми на рівні окремо взятих модулів, функцій чи класів. Мета модульного тестування полягає у виявленні локалізованих у модулі помилок у реалізації алгоритмів, а також у визначенні ступеня готовності системи до переходу на наступний рівень розробки та тестування. Модульне тестування проводиться за принципом «білої скриньки», тобто ґрунтується на знанні внутрішньої структури програми, і часто включає ті чи інші методи аналізу покриття коду.

Модульне тестування зазвичай передбачає створення навколо кожного модуля певного середовища, що включає заглушки для всіх інтерфейсів модуля, що тестується. Деякі з них можуть використовуватися для подачі вхідних значень, інші для аналізу результатів, присутність третіх може бути продиктована вимогами компілятором і збирачем.

На рівні модульного тестування найпростіше виявити дефекти, пов'язані з алгоритмічними помилками та помилками кодування алгоритмів, типу роботи з умовами та лічильниками циклів, а також з використанням локальних змінних

та ресурсів. Було протестовано працездатність кожного модуля програмного продукту.

Тестування на стадії застосування.

На стадії застосування проводилося тестування інтерфейсу готового програмного забезпечення, і тестування як реального часу, яке проводилося користувачами разом із розробниками.

У процесі тестування в реальному масштабі часу перевіряються результати обробки вихідних даних з урахуванням часу їх надходження, тривалості та пріоритетності обробки, динаміки використання пам'яті та взаємодії з іншими програмами.

Тестування програмного продукту проводилося разом із кінцевими користувачами. При тестуванні були виявлені певні недоліки інтерфейсу та програмні помилки.

Програмні помилки були оброблені та справлені. Помилки інтерфейсу було виправлено з урахуванням зауважень користувачів. Внаслідок хорошої роботи тестерів було виявлено майже всі помилки коду, невиявлені помилки найближчим часом будуть виявлені та виправлені.

У зв'язку з термінами, в які необхідно було розробити програмний продукт та обсягу роботи, можна поставити високу оцінку програмного продукту. На цьому наголосили як сам автор, так і команда.

Таким чином, програмний продукт можна вважати завершеним та впроваджувати його на хостинг.

ВИСНОВКИ ДО РОЗДІЛУ 3

У розділі розроблено архітектуру прототипу. У ході вирішення цієї задачі були детально розроблені основні компоненти архітектури. Створено процес їхньої взаємодії. Архітектура розроблялася з використанням найактуальніших та передових на момент написання дипломної роботи технологій.

Також розробка велася з урахуванням обмежень кросплатформенності та розширюваності.

Реалізовано архітектуру прототипу за допомогою таких технологій JavaScript, HTML, CSS.

Проведено тестування. В результаті чого клієнтська та серверна програми повністю "покриті" тестами, це дозволяє вносити зміни або додавати нову функціональність без ризику, що функції, що раніше працюють, виявляться непрацездатними.

Успішно проведено "ручне" тестування.

ВИСНОВКИ

Автосалон – це місце продажу, зберігання та обслуговування автомобілів. І він має місце в нашому сучасному суспільстві виходячи з того, що автомобільний транспорт займає важливе місце в єдиній транспортній системі країни. Він перевозить понад 80% народногосподарських вантажів, що з високою маневреністю автомобільного транспорту, можливістю доставки вантажів «від дверей до дверей» без додаткових перевантажень у дорозі, отже, високої швидкості доставки і збереженням вантажів.

Велика протяжність автомобільних доріг забезпечує можливість їхньої повсюдної експлуатації при значній провізній спроможності. Висока мобільність, здатність оперативно реагувати на зміни пасажиропотоків ставлять автомобільний транспорт "поза конкуренцією" при організації місцевих перевезень пасажирів. На його частку припадає майже половина пасажирообігу.

Автомобільний транспорт зіграв величезну роль формуванні сучасного характеру розселення людей, поширенні далекого туризму, в територіальній децентралізації промисловості та середовища обслуговування.

Великий вибір автомобілів, що пропонуються до продажу, необхідність зберігання даних про велику кількість виробників, постачальників та покупців ускладнюють завдання обліку постачання, продажу та залишків автомобілів в автомагазині. Ручна обробка цих даних не дозволяє отримати оперативну інформацію про зменшення кількості автомобілів, які мають попит і, отже, необхідність поповнення цієї кількості автомобілів. Розрахунок прибутку, витрат, рентабельності, пошук рішень, що призводять до підвищення ефективності автомагазину, стають трудомісткими.

У будь-якій організації, як великої, так і маленької, виникає проблема такої організації управління даними, яка забезпечила б найбільш ефективну

роботу. Деякі організації використовують шафи для зберігання папок з документами та ручний пошук та обробку необхідної інформації. Але більшість воліють комп'ютеризовані бази даних та програми, що дозволяють ефективно зберігати, швидко отримувати потрібну інформацію та керувати великими обсягами даних.

Велика кількість інформації, високі вимоги до точності, численні обчислення, потреба у постійному оновленні даних роблять необхідним застосування баз даних для обліку автомобілів в автомагазині.

Об'єкт автоматизації - організація замовлення автомобілів продажу через автомагазин з метою отримання прибутку.

Предмет автоматизації – облік залишків автомобілів в автомагазині для забезпечення своєчасного замовлення автомобілів, запас яких може закінчитися найближчим часом.

Мета розробки інформаційної системи – збільшення прибутку за рахунок своєчасності замовлень на постачання автомобілів в автомагазин та зниження трудомісткості облікових операцій.

Під час виконання бакалаврської роботи було створено дизайн та розроблено веб портал автосалону. Проаналізовано сучасні технології у створенні веб-сайтів. Найпопулярнішою тенденцією останніх років є мінімалістичний дизайн, який виключає нагромадження об'ємних об'єктів. Проаналізовано існуючі засоби для розробки веб-сайтів: порівняння серверних мов програмування PHP, Ruby, Python, різноманітних середовищ розробки та легких редакторів. Розроблено мережну структуру веб-сайту. Основні розділи веб-сайту: Контакти, Каталог, Ціни, Доставка и оплата, Про компанію . Розроблено дизайн веб-сайту згідно з основними принципами мінімалістичного дизайну. Створено меню сайту. Розроблено горизонтальне меню каталогу та головне верхнє горизонтальне меню. На детальній сторінці виводиться форма

замовлення виробу. Протестовано веб-сайт у найпопулярніших браузерях та була проведена перевірка валідації даних. В ході тестування виявлено, що сайт працює коректно. Для готового програмного продукту було проведене повноцінне тестування: перевірка гіперпосилань, перевірка відповідності текстів заданій темі, перевірка відповідності сайту індивідуальному завданню, перевірка на кросбраузерність та юзабіліті. Отже, в ході виконання бакалаврської роботи було продемонстровано практичні навички створення та розробки веб-сайту.

1. Здійснена робота дозволила створити сайт для автосалону.

Впровадження нових технологій дозволяє проводити подальші розробки з метою покращення якості та розширення її можливостей та в сукупності з модульною організацією легко реалізовувати їх програмно.

2. За допомогою цього програмного продукту інформаційна система виходить на рівень web-технологій та дозволяє створити представництво організації в Інтернеті. Web-сайт дозволить забезпечити тіснішу взаємодію з користувачами через Інтернет. За допомогою даного програмного продукту клієнти мають можливість: отримати потрібну інформацію не виходячи з дому, а також поспілкуватись з іншими автолюбителями.

3. Розроблений відображає вимоги користувачів, має помірні потреби в апаратних ресурсах, заснований на платформеннонезалежній технології.

4. Web-сайт пройшов ряд тестових випробувань щодо виявлення помилок у проектуванні та реалізації програми. Програмний продукт відповідає всім вимогам ГОСТу і має функціональність, надійність, мобільність і зручність використання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Орлов Л. А. Як створити електронний магазин в Інтернет / Л. А. Орлов. – М.: БУК-ПРЕС, 2006. – 384 с.
2. Калужский М.Л. Электронная коммерция / М. Л. Калужский. – М.: Экономика, 2014. – 328 с.
3. Старовойтова Т. Ф. Электронный бизнес и коммерция / Т. Ф. Старовойтова. – К.: ТетраСистемс, 2009. – 144 с.
4. Нельзина О. Г. Информационные системы для электронной коммерции. Учебное пособие / О. Г. Нельзина. – СПб.: Феникс, 2008. – 265 с.
8. Технологии создания сайтов [Электронный ресурс] // Режим доступа : http://www.dbest.ru/information/advice/site_technologies/
9. Лабберс П. HTML5 для профессионалов. Мощные инструменты для разработки современных веб-приложений / П. Лабберс. – М.: Вильямс, 2011. – 272 с.
10. Ши Д. Философия CSS дизайна / Д. Ши. – СПб.: НТ Пресс, 2015. – 312 с.
11. Зандстра М. А. Объекты, шаблоны и методики программирования, 3-е издание / М. А. Зандстра. – СПб.: Издательский дом «Питер», 2011. – 560 с.
12. Буч Г. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, А. Якобсон. – М.: ДМК Пресс, 2007. – 496 с.
13. Кэмпбел М. Строим Web-сайты. Дизайн. HTML. CSS / М. Кэмпбел. – М.: Триумф, 2006. – 480 с.
14. Джамса К. Эффективный самоучитель по креативному Web-дизайну / К. Джамса. – СПб.: ДиаСофтЮп, 2005. – 672 с. 58
15. 5 причин использовать минималистический дизайн [Электронный ресурс] // Режим доступа : <http://www.imagecms.net/blog/obzory/minimalisticheskii-dizain-saita>
16. Фрейн Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Бен Фрейн – СПб.: Питер, 2014. – 304 с.

17. Раскин Д. Интерфейс: Новые направления в проектировании компьютерных систем / Д. Раскин. – М.: Символ-Плюс, 2004. – 272 с.
18. Когзолл Д. PHP 5. Полное руководство / Д. Когзолл. – М.: Диалектика, 2006. – 752 с.
19. PHP, Ruby, Python – краткая характеристика языков программирования [Электронный ресурс] // Режим доступа :
http://www.internettechnologies.ru/articles/article_1991.html
20. Саммерфилд М. Программирование на Python 3 / М. Саммерфилд. – СПб.: Символ-Плюс, 2009. – 608 с.
21. Инструменты и библиотеки для разработчика приложений PHP [Электронный ресурс] : // Режим доступа : <http://kb.mista.ru/article.php?id=670> –
22. Сравнение сред разработки PHP [Электронный ресурс] : // Режим доступа: <https://toster.ru/q/8531>
23. Уиттакер Д. А, Арбон Д. В. Как тестируют в гугл / Д. А Уиттакер, Д. В. Арбон. – СПб.: Питер, 2014. – 385 с.
24. Стотлемайер Д. Тестирование Web-приложений / Д. Стотлемайер – М.: Кудиц-образ, 2003. – 240 с.
25. Максимчук Р.А., Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений. Издательство: Вильямс, 2008г.
19. Дэвид Макфарланд. Большая книга CSS. 2-е издание. Издательство: Питер, 2011г.
20. Хаген Граф Создание веб-сайтов с помощью Joomla 1.5. Издательство: Вильямс 2009г.
21. Лэнс Лавдей. Проектирование красивых прибыльных веб-сайтов. Издательство: Манн, Иванов и Фербер 2011г.
22. И. И. Попов, Максимов Н.В., Голицына О.Л. Базы данных. Издательство: Форум ИНФРА-М, 2011 г.
23. Лерми П., Фаро С. Рефакторинг SQL-приложений. Издательство: Символ-Плюс, 2009 г.

24. Пейтон К., Меллер. А PHP 5 и MySQL 5 в примерах и на проектах. Издательство: Бином, 2009 г.

25 Пер. с англ./Лаура Томсон, Люк Веллинг. «Разработка web-приложений на PHP и MySQL»:.. – 2 изд., испр. – СПб: ООО «ДиаСофтЮП», 2003г.

26. А. Качанов, В. Ткаченко, А. Головин «Букварь по PHP и MySQL», версия 1.1, 2000г.

+27. Кирсанов Д. Веб-дизайн. И

здательство: Символ-Плюс серия: Библиотека дизайнера, дата выхода: январь 2001

28. Вирин Ф. Интернет-маркетинг. Полный сборник практических инструментов. Издательство: Эксмо, дата выхода: сентябрь 2010

29. С. А. Христочевский. Электронные мультимедийные учебники и энциклопедии. Информатика и образование, 2000г.

30. Кронан Д., Мэтьюз М. Динамическое веб-программирование. Издательство "ЭКСМО" · 2010 г.

31. Л.Х. Зайнутдинова. Создание и применение электронных учебников. Астрахань, ООО "ЦНТЭП", 1999г.

32. Н. Д. Изергин, А. А. Кудряшов, А. Ю. Руднев, В. А. Тегин. «Разработка электронных учебных изданий», учебно-практическое пособие, 2005.

33. Скотт Хокинс, пер. с англ. и редакц.: Н. В. Воронина. «Администрирование web – сервера Apache и руководство по электронной коммерции», издательский дом «Вильяме», Москва, 2001г.