

ВСТУП

Актуальність теми дослідження обумовлена необхідністю аналізу існуючих методів розробки корпоративних і інформаційних ресурсів для виявлення найбільш ефективного методу їх створення, визначення ефективних методів розробки для грамотної побудови корпоративних і інформаційних Інтернет-ресурсів. Під методами створення інтернет-сайтів розуміється сукупність прийомів і інструментів розробки.

Всі методи розробки web-сайтів можна умовно поділити на дві групи. До першої групи належать методи створення сайтів ручного написання з використанням мови розмітки HTML і різних мов програмування. Для написання статичного сайту досить поєднання HTML і CSS. Але якщо необхідний більш складний, динамічний сайт, то тут підключають мови програмування. Основними мовами програмування, які найбільш часто використовуються для розробки сайтів, є PHP, Java script, APS.NET. Друга група методів розробки сайтів простіша по виконанню. Так як вона не вимагає широких знань в області програмування. Розробнику сайту надається можливість через графічний інтерфейс спеціальних програм або CMS-систем створювати сайти.

Управління базами даних пройшло більше чотирьох десятиліть еволюції, забезпечуючи широкий спектр досліджень та широкий спектр технологічних рішень. Спільнота дослідників баз даних та індустрія програмного забезпечення реагували на численні виклики, спричинені змінами у вимогах користувачів та можливостях, які надає апаратний прогрес. Реляційний підхід до баз даних, представлений базами даних SQL, був особливо успішним і однією з найбільш довговічних парадигм у обчислювальній техніці. Найновіші проблеми з базами даних включають бази даних в Інтернеті - бази даних, які управляють сотнями мільйонів користувачів, та хмарні бази даних, які використовують нові методи управління великими обсягами даних.

Бази даних, зокрема реляційні бази даних, є повсюдною частиною

сучасного обчислювального середовища. Системи управління базами даних підтримують найрізноманітніші програми, від ділових до наукових та останнім часом різні типи програм для Інтернету та електронної комерції. Системи управління базами даних (СУБД) сьогодні є основною технологією в більшості організацій і використовують критично важливі програми, на які покладаються банки, лікарні, авіакомпанії та більшість інших типів організацій для своєї повсякденної діяльності. Протягом останніх трьох десятиліть технологія реляційних СУБД виявилася надзвичайно пристосованою і еволюціонувала, щоб задовольнити нові вимоги до програми та постійно зростаючий обсяг і складність даних.

Метою даної роботи є створення веб-ресурсу для ТОВ «Трансфрайт» з базами даних.

Об'єкт роботи – дані логістичної компанії ТОВ «Трансфрайт».

Предмет роботи – веб-сайт з базами даних.

Розробка такої системи є актуальною в цей час, тому що це підвищує ефективність діяльності підприємства й полегшує роботу користувача.

Дана система створюється для ТОВ «Трансфрайт» за допомогою мови програмування PHP, каскадних таблиць стилів (CSS) та мови розмітки HTML. В результаті веб-ресурс буде мати детальну інформацію про підприємство, клієнтів, перевізників та перевезення компанії.

1 ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ АВТОМАТИЗОВАНИХ СИСТЕМ З МАСИВАМИ ДАНИХ ДЛЯ ПІДПРИЄМСТВ

1.1 Огляд літератури з проблематики дослідження автоматизованих систем з масивами даних для підприємств

На сьогоднішній день не існує точного визначення об'єктно-орієнтованого програмування (ООП) або об'єктно-орієнтованої мови програмування. У літературі різні автори дають різне пояснення цих термінів [5] [8] [16]. Виходячи з цих визначень, вони визначають об'єктно-орієнтовану мову програмування як мову програмування, базовими елементами є об'єкти, що мають власні атрибути та методи, і формують ієрархічно організовані класи об'єктів.

Об'єкт - це модель (абстракція) реальної сутності в системі програмування.

Клас - це абстрактне оголошення атрибутів і методів для групи подібних об'єктів, які називаються екземплярами класу.

Атрибут - це параметр, оголошений у класі, який характеризує об'єкт (екземпляр класу).

Метод оголошений у процедурі класу, яка визначає поведінку екземплярів класу.

Загалом, об'єктно-орієнтований підхід до розробки програм, заснованих на чотирьох основних механізмах: абстракція, інкапсуляція, поліморфізм та успадкування.

Абстракція - це процес ідентифікації суттєвих характеристик об'єкта, які відрізняють його від усіх інших видів об'єктів і, таким чином, забезпечуючи чітко визначені концептуальні межі, з точки зору спостерігача.

Інкапсуляція - це процес розділення елементів абстракції, що становлять її структуру та поведінку.

Поліморфізм - це здатність призначати різне значення або використання у різних контекстах, а властивість об'єкта реагувати на запит відповідно до його типу.

Спадкування - це механізм оголошення нових типів даних на основі існуючих типів таким чином, що атрибути та методи базових типів стають членами підтипу [10] .

Орієнтація на об'єкти в PHP була повністю підтримана у версії 5.3, але частково підтримувалась і використовувалася за кілька років до цього, дивлячись на ранні версії 4.x. Тож був період, коли процедурні системи могли мігрувати код до класів [9].

Розглянемо основні проекти з відкритим кодом, реалізованими в PHP:

Wordpress: Найпопулярніше програмне забезпечення для ведення блогу; він має величезну спільноту як авторів, так і активних користувачів.

Drupal. Один з найдосконаліших CMS. Він також характеризується великою і активною спільнотою.

PhpBB: одне з найбільш широко використовуваних програмних засобів форумів.

MantisBt: Мабуть, найпопулярніший додаток для відстеження помилок, написаний на PHP.

PhpMyAdmin: Добре відомий інструмент адміністрування MySQL.

PHP є досить новою мовою програмування, і набула популярності протягом останнього десятиліття. Непрямим показником зрілості для даної мови програмування є розвиток сторонніх бібліотек та їх. Тенденцією розвитку сучасного програмного забезпечення є використання таких готових будівельних блоків для полегшення та пришвидшення розробки додатків. Побічним ефектом є введення невикористаних блоків коду через сценарій мови. Однак той факт, що вихідний код бібліотеки стає частиною вихідного коду системи, дозволяє виміряти співвідношення бібліотечного коду до системного коду, що не просто з компільованими мовами.

У мовах сценаріїв основним джерелом невикористаного коду є використання сторонніх бібліотек, що одночасно є загальновизнаною передовою практикою у розробці програмного забезпечення та можливим свідченням зрілості [11]. У цьому контексті дослідили кількість бібліотечного коду, який

використовувався з часом у кожній системі. Іншим фактором, що передбачає зрілість програмного забезпечення, є стабільність відповідних API, і тому було розглянуто шість класів можливих змін API.

На рис.1.1 представлено співвідношення кількості методів до загальної кількості функцій та методів класів системного коду, виключаючи сторонні бібліотеки, щоб показати тенденцію перетворення основної кодової бази систем у класи. Відзначимо, що Drupal після тривалого періоду заперечення об'єктно-орієнтованої парадигми, навіть усуваючи незначну частину класів, що існували в ранніх версіях, зробив поворот у версії 7.0 із введенням класів. Проект із основною зміною своєї парадигми кодування - phpBB, де у версії 3.0.0, що стала віхою в історії проекту, він масово прийняв об'єктну орієнтацію.

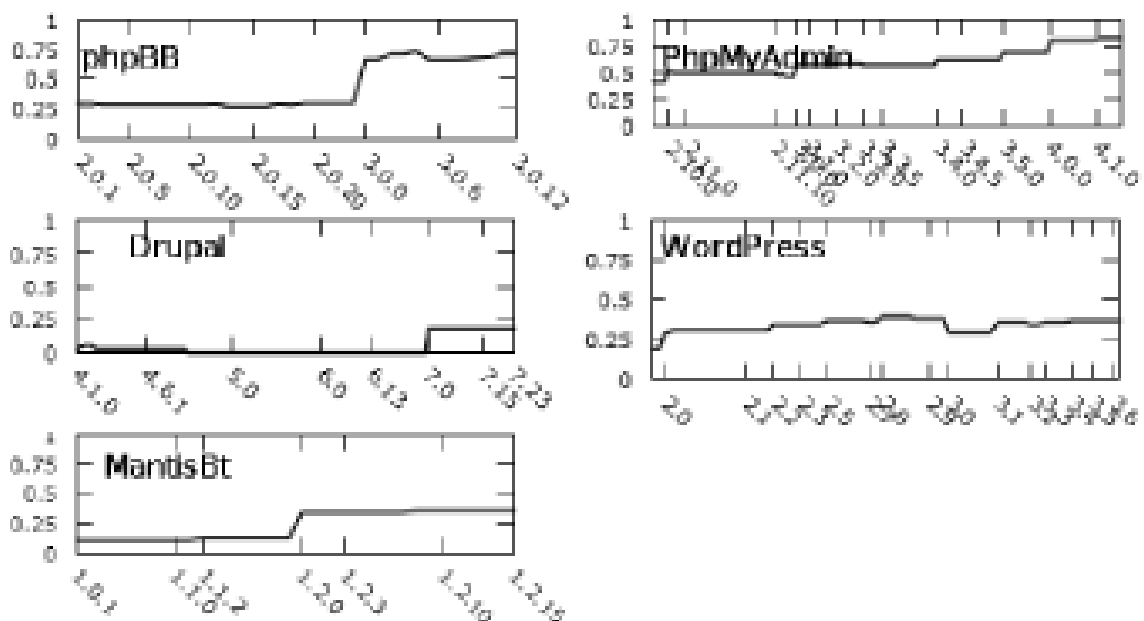


Рисунок 1.1 – Співвідношення методів до загальної кількості функцій та методів

WordPress зберігає повільну, але стійку тенденцію до орієнтації об'єктів, але величезний користувальницький код у плагінах та темах утримує команду розробників від основних перезаписів у загальнодоступний API програми. Натомість розробники поступово виконують рефакторинг додатків до внутрішніх

систем, не порушуючи зворотної сумісності. З іншого боку, широко використовуваний проект phpMyAdmin, який можна знайти майже на будь-якому веб-сервері під управлінням Linux, має потужний імпульс до того, щоб бути повністю об'єктно-орієнтованою системою. Це пов'язано з мінімальною кількістю користувацьких плагінів, що не спричиняє загрози для порушення загальнодоступного API програми.

Таблиця 1.1 – Огляд PHP додатків

Тема	Висновки	Wordpress	phpBB	Drupal	MantisBt	PhpMy Admin
Зрілість програмного забезпечення (використання бібліотеки)	Проекти повторно використовують код, що включає сторонні бібліотеки	ТАК	ТАК	НІ	ТАК	ТАК
Зрілість програмного забезпечення (стабільність інтерфейсу)	Інтерфейс функції залишається стабільним	ТАК	ТАК	ТАК	ТАК	ТАК
Якість програмного забезпечення	Складність залишається стабільною	ТАК	ТАК	ТАК	ТАК	ТАК
Прийняття об'єктно-орієнтованого програмування (ООП)	Проекти поступово переходять на ООП	ТАК	ТАК	ТАК	ТАК	ТАК
Безпека	Вирішення проблем уразливості, наприклад Введення SQL, міжсайтовий сценарій, автоматична реєстрація	ТАК	ТАК	ТАК	НІ	НІ

Перехід додатків із процедурної на об'єктно-орієнтовану парадигму - це не лише питання волі розробника чи мови реалізації, але й те, чи може проект дозволити собі витрати на розрив зворотної сумісності, що накладає значні

проблеми на їхніх клієнтів.

Огляд розглянутої теми з одиницею аналізу та висновки, виведені на основі висновків для кожного проекту, наведені в таблиці ТАК, означає, що отриманий висновок можна вважати підтвердженням для відповідного проекту, тоді як НІ означає, що висновок є не підтверджено.

Сіаме та Кунда у своїй роботі розглядаються деякі дослідницькі роботи, здійснені в еволюції програм РНР, які існували і широко використовуються [15]. В даний час РНР є однією з найпопулярніших мов програмування, яка широко використовується як у спільноті з відкритим кодом, так і в промисловості для створення великих веб-орієнтованих програм та фреймворків. У своєму дослідженні вони розглядають розвиток РНР-програм з точки зору використання бібліотек, зрілості програмного забезпечення, прийняття об'єктно-орієнтованої парадигми, еволюції складності та безпеки. Результати свідчать про те, що ці системи проходять систематичне обслуговування, а еволюція допомагає зростати основній мові програмування.

Між концепціями бізнес-аналітики на найвищому рівні та дрібними інструкціями щодо використання інструментів постачальників лежить важливий, але недостатньо вивчений рівень архітектури, дизайну та процесу. Без цих знань великі дані применшуються – проекти не здійснюються, затримуються і перевищують бюджет. «Посібник з бізнес-аналітики: від інтеграції даних до аналітики» яскраво висвітлює часто знехтувану тему, озброюючи знаннями, необхідними для розробки надійних процесів бізнес-аналітики та інтеграції даних. Практикуючий професор з бізнес-аналітики (Business Intelligence – BI), Р. Шерман, робить висновки щодо створення систем, які є економічно ефективними, багаторазовими та необхідними для перетворення необроблених даних у цінну інформацію для тих, хто приймає рішення у бізнесі [23].

Ключові особливості його дослідження:

- 1) Надає практичні вказівки щодо створення успішних рішень для інтеграції BI та даних.
- 2) Пояснює основи BI та інтеграцію даних, архітектуру та процеси

чіткою доступною мовою.

3) Включає повний життєвий цикл розробки проектів, який можна застосовувати як на великих підприємствах, так і на малому та середньому бізнесі.

4) Описує найкращі практики та прагматичні підходи, щоб програмісти могли застосувати їх до дії.

5) Супутній веб-сайт містить шаблони та приклади, подальше обговорення ключових тем, матеріали для викладачів та посилання на надійні галузеві джерела.

1.2 Аналіз сучасних інформаційних систем у сфері транспорту

Розглянемо сучасне програмне забезпечення для ведення діяльності та обліку в сфері вантажоперевезень і логістики. Програмні комплекси спеціально розроблені для експедиторів, перевізників, диспетчерів і дозволяють вести документообіг і бухгалтерський облік в транспортних компаніях.

Програмний комплекс «Мегалогіст» розроблений на платформі «1С: Підприємство 8». Призначений для комплексної автоматизації транспортної логістики. Програма дозволяє створювати завдання на перевезення, а також планувати маршрут в ручному і автоматичному режимі, контролювати виконання рейсів в онлайн-режимі, проводити аналіз KPI і рентабельності доставки.

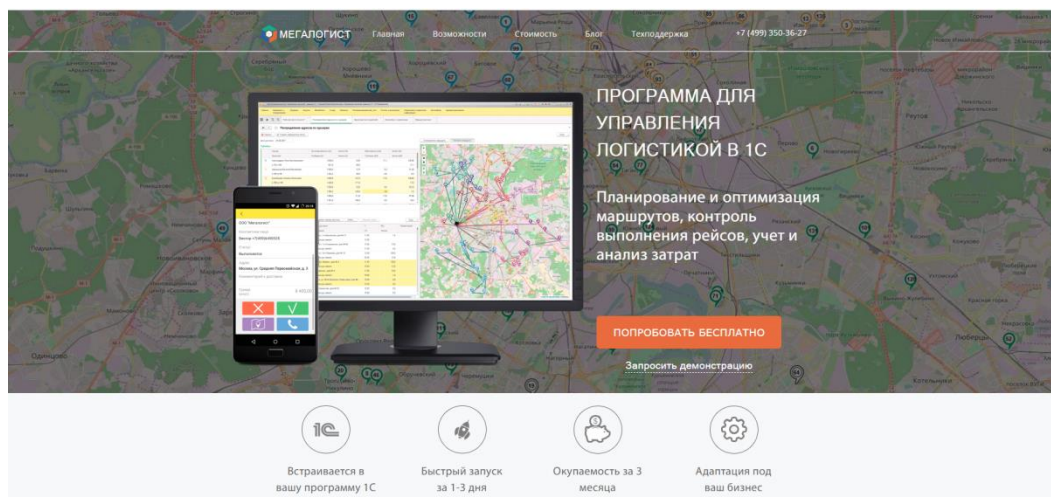


Рисунок 1.2 – Сайт: <https://mega-logist.ru/>

Є мобільний додаток для водіїв і кур'єрів.

Тарифи:

Вартість безлімітної ліцензії на програму 210000 руб. Ліцензія оплачується один раз і не має обмеження на кількість користувачів, кількість замовлень і термін дії.

Якщо у підприємства менше ніж 7500 замовлень на місяць, тобто є можливість придбати тариф з обмеженням кількості оброблюваних замовлень в місяць (одноразова оплата). Можливий апгрейд з будь-якого тарифу.

- 1) Тариф «500» - 19250 грн.
- 2) Тариф «2500» - 32700 грн.
- 3) Тариф «5000» - 48100 грн.
- 4) Тариф «7500» - 71200 грн..

Є повнофункціональна демо-версія на 14 днів. Безкоштовна віддалена презентація та відповіді на питання.

Програма «Вантажоперевезення» дозволяє приймати і контролювати заявки на перевезення, зберігання, перевалку, завантаження, розвантаження, страхування тощо послуги для стандартних, збірних, негабаритних, мультимодальних та ін. видів вантажів. Є тарифні сітки, облік гсм, складський облік, планування і контроль ремонтів т.с., розрахунок зарплати, формування всіх видів документації, а також різних видів звітів, аналіз і контроль боргів, розрахунок рентабельності по

транспорту, менеджерам і всьому підприємству в цілому. Відправка e-mail, смс, вбудована перевірка контрагентів за багатьма параметрами в системі «Контур.фокус».

О ПРОГРАММЕ

ЦЕНА

КОНТАКТЫ

СКАЧАТЬ ДЕМО

+7 (919) 282-62-05

Программа "Грузоперевозки"

Программа "Грузоперевозки" предназначена для автоматизации работы транспортных компаний любого типа.

Лучшая цена на рынке! 5.800 рублей за одно рабочее место разовая оплата, без каких-либо доплат в дальнейшем.

Купить программу




Рисунок 1.3 – Сайт: <https://программагрузоперевозки.рф>

Якщо клієнт хоче, щоб програма працювала через інтернет, потрібно звернутись до розробників. Безкоштовно проконсультують як налаштувати програму Вантажоперевезення для віддаленої роботи через Інтернет.

Тарифи:

Купити програму Вантажоперевезення з одноразовою оплатою на необмежений час:

- 1) На 1 робоче місце - 2200 грн.;
- 2) На 5 робочих місць - 8800 грн.;
- 3) На 10 робочих місць - 15400 грн.;
- 4) Ліцензія без обмеження кількості робочих місць - 23100 грн.

Орендувати програму Вантажоперевезення на 6 місяців:

- 1) На 1 робоче місце - 330 грн.;
- 2) На 5 робочих місць - 1900 грн.;
- 3) Ліцензія на 10 робочих місць - 3300 грн.;
- 4) Ліцензія без обмеження кількості робочих місць - 7700 грн.

Купити програму Вантажоперевезення в розстрочку на 1 рік без додаткової націнки:

- 1) Ліцензія на 1 робоче місце - 3300 грн.;
- 2) Ліцензія на 5 робочих місць - 1900 грн.;
- 3) Ліцензія на 10 робочих місць - 3300 грн.;

Ліцензія без обмеження кількості робочих місць - 7700 грн.

Демоверсія з періодом 30 днів. Відповіді на будь-які питання по програмі.
Безкоштовна віддалена презентація. Допомога в установці та налаштуванні.

Програма АВТОПЛАН для автоматизації контролю, обліку і управління на автотранспортних підприємствах і транспортних підрозділах. Рентабельність від 10 одиниць автотранспорту / спецтехніки. Повний контроль транспорту, ПММ, заявок, персоналу, ТМЦ, термінів, процесів. Автоматизація управління, інтеграція GPS / ГЛОНАСС, розумний контроль даних, економія до 20% всіх витрат підприємства, різке зниження ручної роботи до 40%, виняток помилок персоналу і фальсифікацій даних.

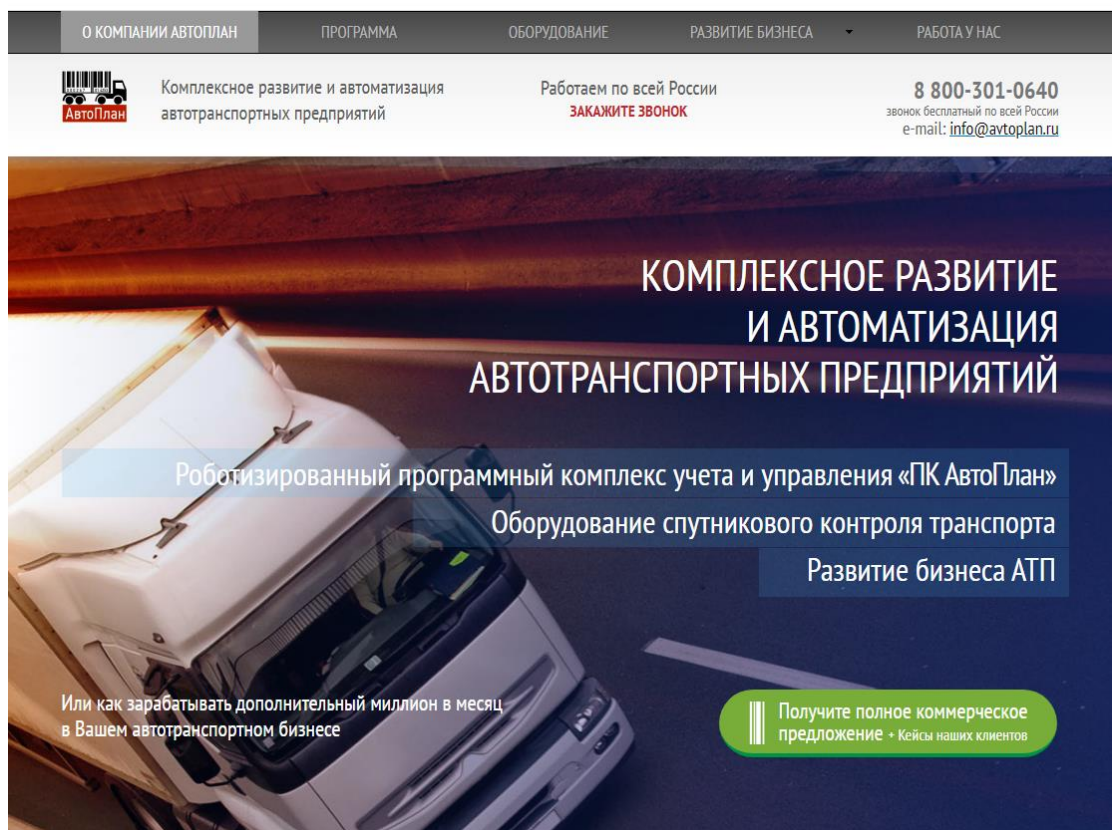


Рисунок 1.4 – Сайт: <http://avtoplan.ru/>

Комплексна автоматизація підприємства: програмний комплекс,

впровадження, навчання, доробка під замовника, постійна підтримка, оновлення.

Тарифи:

- 1) Тариф «Стартовий» - 1900 грн. в місяць (2 робочих місця);
- 2) Тариф «Базовий» - 3300 грн. в місяць (10 робочих місць);
- 3) Тариф «Високий рівень» - 5700 грн. в місяць (15 робочих місць);
- 4) Тариф «Договірний» - за запитом;
- 5) Довічні ліцензії з одноразовою оплатою - за запитом.

Є тестовий період 1 місяць.

Інформаційна система «Kiberlog» дозволяє управляти основними бізнес-процесами в сфері вантажоперевезень, забезпечувати своєчасний обмін інформацією та вести правильний документообіг з замовниками і вантажоперевізниками.



Программа для транспортных компаний и владельцев транспорта

Формируйте, печатайте Заявки, бухгалтерские документы, отправляйте документы по электронной почте в один клик, контролируйте перевозки и финансы, управляйте сетью ваших филиалов, создавайте личные кабинеты Заказчикам и Перевозчикам.

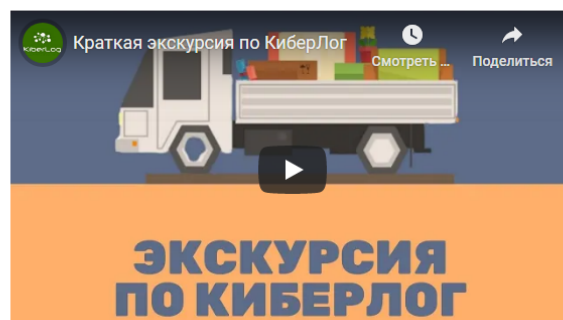


Рисунок 1.5 – Сайт: www.kiberlog.ru

Тарифи:

Розробники Kiberlog пропонують 3 тарифу: «стандарт», «організація» і «корпоративний»:

- 1) Тариф «стандарт» - 170 грн. в місяць за одного користувача (до 4-х користувачів);
- 2) Тариф «організація» - 160 грн. в місяць за одного користувача (до 20-і користувачів);
- 3) Тариф «корпоративний» - 150 грн. в місяць за одного користувача (з вище 20 користувачів).

Є тестовий період: 1 місяць

Як ми бачимо, широкий спектр програм для логістики представлений російськомовним контентом. Ці програми оплачуються щомісяця, тому розробка власної бази даних для підприємства ТОВ «Трансфрайт» є актуальною.

2 МЕТОДИ ТА ВИБІР МЕТОДИКИ РОЗРОБКИ ВЕБ-РЕСУРСУ З БАЗАМИ ДАНИХ

2.1 Постановка задачі

Спроекувати і розробити сайт для ТОВ «Трансфрайт». Звичайні користувачі зможуть переглядати основну інформацію про ТОВ «Трансфрайт». ТОВ «Трансфрайт» повинен мати сайт на якому буде меню, що складається з («Компанія», «Клієнти», «Перевезення», «Перевізники» і «Угоди»).

Веб-сайт ТОВ «Трансфрайт» буде виглядати наступним чином. На головній сторінці буде розташовано меню для швидкого переходу по веб-сайту. Меню буде складатися з:

- Компанія;
- Клієнти;
- Перевезення
- Перевізники;
- Угоди.

Перший пункт меню «Компанія» в цьому пункті буде відображатися основна інформація про ТОВ «Трансфрайт».

Другий пункт меню «Клієнти» в цьому пункті буде відображатися основна інформація про клієнтів компанії ТОВ «Трансфрайт».

Третій пункт меню «Перевезення» в цьому пункті буде відображатися вибіркова інформація про перевезення.

Четвертий пункт меню «Перевізники» в цьому пункті будуть відображатися дані вантажівок.

П'ятий пункт меню «Угоди» в цьому пункті будуть відображатися дані угод, а саме: дата, клієнт, перевізник.

2.2 Механізм проектування сайту та бази даних

Об'єктно-орієнтований підхід є якісним, продуктивним та гнучким. Інформаційні системи, розроблені за традиційним підходом, як відомо, схильні до помилок, дорогі та негнучкі. Об'єктно-орієнтований підхід може зменшити помилки, витрати та збільшити гнучкість завдяки своїм властивостям.

Під час проектування сайту важливим є обсяг діяльності, який можна узагальнити наступним чином:

- 1) Встановлення вимог передбачає консультації та узгодження зацікавлених сторін щодо того, чого вони хочуть від системи, що виражається як заява про вимоги.
- 2) Аналіз починається з розгляду заяви про вимоги і закінчується виготовленням специфікації системи. Специфікація - це офіційне подання того, що повинна робити система, виражена термінами, які не залежать від того, як вона може бути реалізована.
- 3) Розробка починається із специфікації системи, розробляється проектна документація та надається детальний опис того, як повинна будуватися система.
- 4) Реалізація - це побудова комп'ютерної системи відповідно до заданої проектною документації та з урахуванням середовища, в якому буде функціонувати система (наприклад, конкретного обладнання або програмного забезпечення, доступного для розробки). Впровадження може здійснюватися поетапно, як правило, з початковою системою, яку можна перевірити та протестувати до того, як остаточна система буде випущена для використання.
- 5) Тестування порівнює впроваджену систему із проектною документацією та специфікацією вимог і створює звіт про прийняття або, як правило, список помилок, які вимагають перевірки процесів аналізу, проектування та впровадження (виправлення, як правило, є завданням, що

повторюється через життєвий цикл).

- б) Технічне обслуговування передбачає вирішення змін у вимогах або середовищі реалізації, виправлення помилок або перенесення системи в нові середовища (наприклад, перенесення системи з автономного ПК на робочу станцію UNIX або мережеве середовище). Оскільки технічне обслуговування передбачає аналіз необхідних змін, проектування рішення, впровадження та тестування цього рішення протягом терміну служби підтримуваної програмної системи, життєвий цикл проекту буде неодноразово переглядатися.

У реляційних базах даних формування зв'язків здійснюється на логічному рівні (тобто за значеннями атрибутів), а в ієрархічних і мережевих БД - на фізичному рівні (за адресами записів).

Заповнення бази даних

Після створення бази даних існує два способи заповнення таблиць - або з наявних даних, або за допомогою користувацьких програм, розроблених для бази даних.

У деяких таблицях можуть бути наявні дані з іншої бази даних або файли даних. Дані можуть також надходити від зовнішнього джерела або створюватися під час завдання введення даних (перетворення паперових копій ручних записів у комп'ютерні файли). У таких ситуаціях найпростішим підходом для заповнення бази даних є використання засобів імпорту та експорту, які містяться в СУБД.

Зазвичай доступні засоби імпорту та експорту даних у різних стандартних форматах (ці функції також відомі в деяких системах як завантаження та розвантаження даних). Імпорт дозволяє копіювати файл даних безпосередньо в таблицю. Коли дані зберігаються у форматі файлу, який не підходить для використання функції імпорту, тоді необхідно підготувати прикладну програму, яка читає старі дані, перетворює їх за необхідності, а потім вставляє в базу даних за допомогою спеціально створеного коду SQL для цього.

Основні завдання проектування бази даних:

- 1) забезпечення зберігання в базі даних всієї необхідної інформації;

- 2) зручна і зрозуміла систематизація інформації включеної в базу даних;
- 3) можливість отримання всієї необхідної інформації по одному запиту користувача;
- 4) скорочення надмірності і повторення (дублювання) інформації;
- 5) забезпечення цілісності бази даних.

MySQL - одна з найпопулярніших баз даних для веб-додатків.

Основні етапи проектування бази даних представлені на рис.2.1.

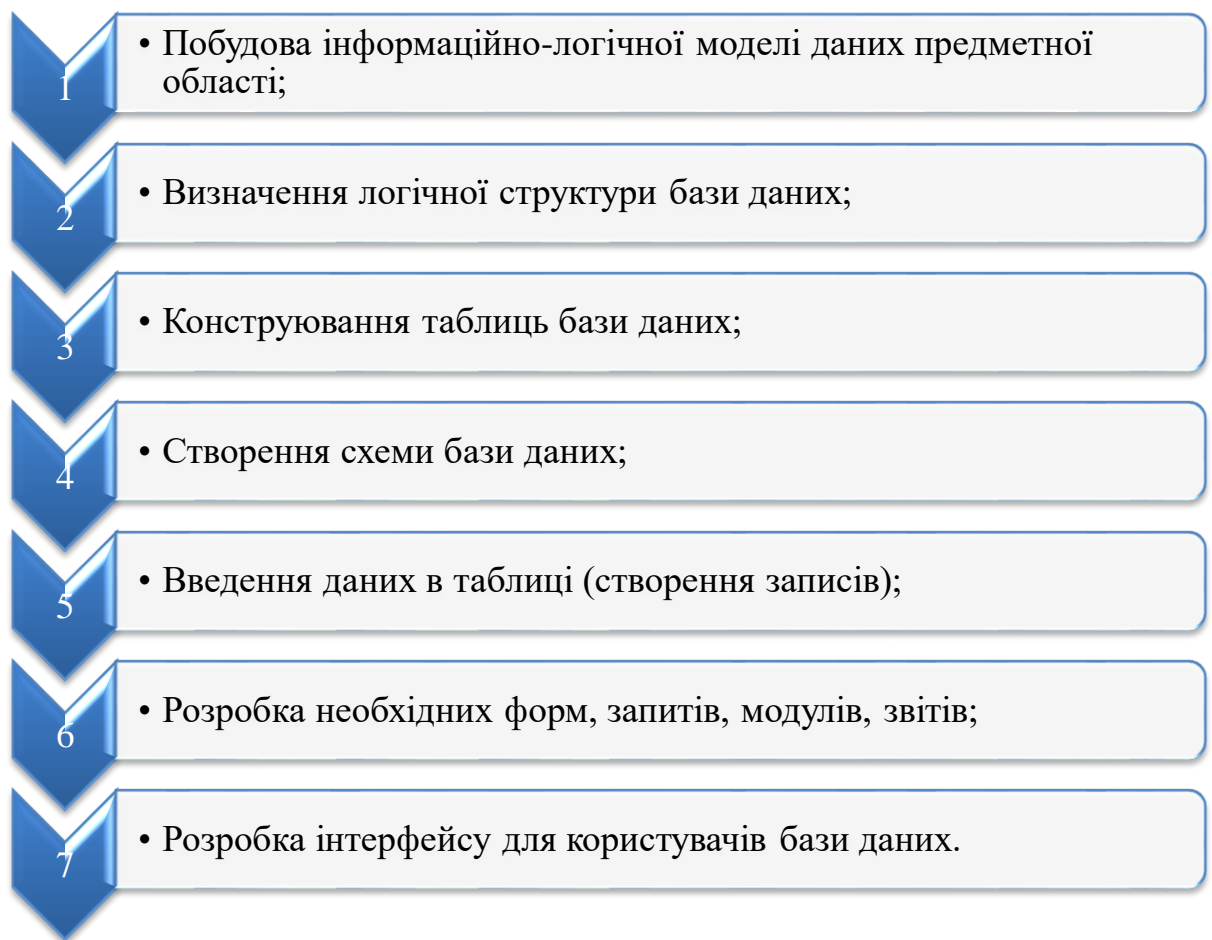


Рисунок 2.1 – Етапи проектування бази даних

Під поетапним процесом проектування бази даних розуміється об'єктивна форма подання і організація сукупності даних.

2.3 Вимоги до інформаційної та програмної сумісності та складу та параметрів технічних систем

Сайт можна переглядати на всіх сучасних операційних системах.

Сучасний комп'ютер є складною системою і складається з багатьох пристроїв введення-виведення інформації (процесор, монітор, диски, клавіатура і т.д.). Управління всіма цими складовими частинами комп'ютера і їх оптимальне використання являє собою досить важке завдання для користувача. Для вирішення цього непростого завдання застосовується спеціальне програмне забезпечення, яке називається операційною системою.

Сучасна операційна система - комплекс програмних засобів, які надають користувачеві стандартне введення-виведення інформації та управління програмами, що спрощують роботу з комп'ютером. Програмний інтерфейс таких операційних систем зменшує розмір конкретної програми і спрощує її роботу. Розглянемо докладніше, що представляють собою сучасні операційні системи: Windows, Linux, UNIX, Mac OS X.

Windows 10. Це операційна система для персональних комп'ютерів, розроблена корпорацією Microsoft після операційної системи Windows 8.1. 29 липня 2015 року корпорація Microsoft офіційно анонсувала свою нову операційну систему Windows 10. Вона стала офіційно доступна для продажу і установки в 190 країнах світу. Перед цією офіційною датою початку продажів пройшло 9 місяців тестування системи. На початку листопада 2015 року було офіційно оголошено про те, що ця операційна система вже встановлена на 110 млн різних пристроях у всьому світі.

UNIX. Операційна система UNIX з'явилася на початку 1970-х рр. У 1975 р почалося масове використання цієї операційної системи. На той час вона була переписана на мові програмування C і це сильно полегшило її переносимість. Так як ця операційна система поширювалася разом з вихідними текстами, вона була першою операційною системою, в яку могли вносити зміни користувачі. Значну роль зіграли відкриті вихідні тексти операційної системи UNIX в поширенні цієї системи в освітній та науковій сферах, в першу чергу в університетах. Вдалі архітектурні рішення UNIX, гнучкість, відкритість і інші функціональні можливості дозволили цій операційній системі завоювати ринок комп'ютерів.

Багато операційних систем, які з'явилися пізніше, включали деякі ключові ідеї UNIX: переносимість на основі мови програмування C, файлову систему, інтерфейс командного рядка та ін.

Операційна система UNIX має два суттєвих плюса: простоту і мобільність. Простота полягає в тому, що ця операційна система завдяки компактності ядра невимоглива до ресурсів комп'ютера. Крім цього, UNIX містить багато інших різних переваг. Наприклад, просту файлову модель, що дозволяє створювати величезну кількість підкаталогів на жорсткому диску. Також використовуються всього 6 основних команд: операція породження `fork`, заміна вбудованих програм на нові програми. Ці системні виклики є простими операціями, з яких і складається операційна система UNIX. Знаючи ці операції, ми можемо виконувати основні операції в UNIX. Значно спростило операційну систему UNIX використання розвиненого командного мови в базовому інтерфейсі. Навіть сьогодні, коли з'явилися численні графічні оболонки, знайдеться багато користувачів, що віддають перевагу перший варіант інтерфейсу командного рядка. Мобільність UNIX полягає в тому, що її можна використовувати на різних апаратних платформах. Крім цього, можна запускати програми відразу декільком користувачам з однієї машини, що полегшує створення мереж. Завдяки цьому операційна система UNIX зіграла велику роль в розвитку мережі Інтернет.

Звичайно, операційна система UNIX неідеальна. Основними її недоліками є: слабка стійкість до апаратних збоїв, низька ефективність при вирішенні завдань одного типу і дуже велика перевантаженість.

Mac OS X. Mac OS - сімейство операційних систем виробництва корпорації Apple. Була розроблена для лінійки персональних комп'ютерів Macintosh.

Сімейство операційних систем OS X є другим за поширеністю (після Windows). Mac OS X відрізняється високою стійкістю, що робить її схожою на попередницю - Mac OS 9. Для створення версії повністю передрукували початковий програмний код. Тепер в Mac OS X використовується ядро Mach, стандартні сервіси BSD (в сукупності підсистема з відкритим кодом Darwin). У ранніх модифікаціях OS X працювала тільки на комп'ютерах з процесорами

PowerPC. Пізніше, починаючи з 10.6, Mac OS X працює тільки з процесорами Intel. Багатозадачність цієї операційної системи дозволяє працювати декільком процесам відразу і не допускати збою всієї системи, якщо стався збій одного з процесів. Найбільш помітно змінився графічний інтерфейс, який в OS X отримав назву Aqua (QuickTime, Quartz Extreme і т.д.). Дуже важливою перевагою Mac OS X є її рівень безпеки під час роботи в Інтернеті, так як вона добре захищена від хакерських атак, а також на даний момент відомо мала кількість вірусів, які можуть її вразити. Завдяки стандартній комплектації, Mac OS X включає безліч технологій, які допомагають людям з обмеженими можливостями використовувати цю операційну систему по максимуму.

Те, що Mac OS не так поширена, як Windows, викликано тим, що принциповим рішенням керівництва компанії Apple є її установка тільки на комп'ютери Macintosh. До того ж ці комп'ютери стоять у багато разів більше, ніж звичайні персональні комп'ютери, тому число користувачів операційної системи Mac OS X порівняно мало.

Linux. В кінці 1980-х рр. з широким розповсюдженням у світі персональних комп'ютерів була створена і реалізована можливість перенести UNIX на персональний комп'ютер. Заслуга повноцінного впровадження UNIX на ПК належить Лінусу Тервальдсу, який в 1991 р опублікував першу версію нової операційної системи, що отримала в його честь назву Linux. Завдяки відкритій публікації і активної участі програмістів різних країн операційна система Linux швидко розвивалася і стала повноцінною і сучасною платформою. Вона дала можливість широкому поширенню ідей та технологій UNIX на персональних комп'ютерах.

Завдяки тому, що вихідний код Linux є відкритим, будь-який програміст можна в будь-який момент додати будь-які доповнення або виправити помилки і недоліки системи. Постійно відбувається вдосконалення системи, розробляються нові версії, випробовуються і впроваджуються найрізноманітніші програми. Відкритість операційної системи Linux дозволяє будь-якому користувачеві не побоюватися шкідливої дії різних вірусних програм.

В операційній системі Linux робота користувачів відбувається через інтерфейс командного рядка (CLI), графічний інтерфейс користувача (GUI) або, в разі вбудованих систем, через елементи управління відповідних апаратних засобів. Linux працює на безлічі процесорів різних архітектур.

Більшість користувачів операційної системи Linux використовують для її установки дистрибутиви (набір рішень і ряд програм для різних завдань, об'єднаних єдиними системами управління, настройки і підтримки). Найбільш поширеними і популярними серед них є Red Hat, UBUNTU і Debian. Кожен з цих дистрибутивів має свою концепцію, свій набір пакетів, свої недоліки і переваги.

Користувач може встановлювати на Linux ті програми і додатки, які йому потрібні. Це позбавляє від необхідності приймати на комп'ютер непотрібні програмні продукти, що займають дисковий простір і не представляють практичної користі.

Не всі компоненти комп'ютера обов'язкові для встановлення, але є перелік компонентів, без яких робота системи неможлива:

- 1) материнська плата;
- 2) процесор (з вбудованим графічним ядром);
- 3) процесорний кулер;
- 4) оперативна пам'ять;
- 5) блок живлення;
- 6) жорсткий диск.

Процесор	<ul style="list-style-type: none"> • Тактова частота 1 ГГц • Архітектура IA-32 або x64 що підтримує PAE, NX та SSE2 • Процесори x64 мають підтримувати PrefetchW та LAHF/SAHF інструкції.
Оперативна пам'ять (RAM)	<ul style="list-style-type: none"> • 4 ГБ
Відеокарта	<ul style="list-style-type: none"> • Графічний пристрій з DirectX 10 WDDM 1.3 або новіший
Екран пристрою	<ul style="list-style-type: none"> • 800×600
Пристрій введення	<ul style="list-style-type: none"> • Клавіатура та мишка
Пам'ять на жорсткому диску	<ul style="list-style-type: none"> • IA-32: 32 ГБ; x64: 32 ГБ
Підключення до Інтернету	<ul style="list-style-type: none"> • Випуски Windows 10, які працюють в S-режимі, потребують підключення до Інтернету під час початкового налаштування пристрою.

Рисунок 2.2 – Рекомендовані апаратні вимоги до ПК

Вибираючи операційну систему для свого пристрою, користувачеві не треба піддаватися впливу моди. Виходити потрібно з тих функцій, для виконання яких і призначена операційна система. Вдалий вибір ОС буде служити гарантією успішної роботи користувача.

3 ДОСЛІДЖЕННЯ РОЗРОБКИ ТА РЕАЛІЗАЦІЇ ПРОГРАМНОГО ПРОДУКТУ

3.1. Проектування програми

Верстка сайту - передбачає формування структури web-сторінки, що розміщає елементи веб-сторінки (зображення, текст і т. д.) у вікні браузера таким чином, щоб елементи дизайну виглядали аналогічно розробленому макету. Верстка веб-сторінок відрізняється від поліграфічної тим, що необхідно враховувати різницю відображення елементів в різних браузерах і різницю в розмірах робочого простору пристроїв.

Процес складний і має творчу основу, жоден із способів не є канонічним і прийнятим як основа. Всі підходи до верстці мають як переваги, так і недоліки. Хоча робота верстальника прихована від очей, саме вона забезпечує безперебійність при роботі на різних пристроях, а також швидкість завантаження кожної сторінки сайту.

Результатом верстки є так званий шаблон сторінки - це зовнішній вигляд сайту, в якому визначається розташування різних елементів на сайті, художній стиль і спосіб відображення сторінок. Використання шаблонів дозволяє проводити гнучку настройку дизайну сайтів, розділів і сторінок сайту.

У сайту може бути багато шаблонів дизайну, і, навпаки, один шаблон може бути використаний на декількох сайтах. Веб-шаблон також є інструментом для відділення вмісту від візуального представлення в веб-дизайні, для масового створення веб-документів.

Для простоти композицію можна визначити, як взаємодія двох або більше елементів в одній графічній області. За допомогою цієї взаємодії автор передає якийсь додатковий сенс, ідею, розставляє акценти.

Сайт з мінімальною структурою або сконцентрований на якомусь головному елементі, або складається з однієї 1-колоночної сторінки. У дизайні

домінує один головний елемент, а весь вміст сторінки підпорядковане йому або відіграє допоміжну роль.

Правильний дизайн сайту повинен відповідати розширенню екрану. Сьогодні найпопулярніше розширення екрана 1280x1024, тому більшість сайтів оптимізують саме під такі параметри. Для кишенькових ПК зазвичай виконують окремі версії сайтів.

Фіксована верстка або статична це традиційний підхід, при якому ширина незмінна при будь-якому дозволі екрану.

Горизонтальне меню у верхній частині сторінки займає менше місця. Лівостороннє вертикальне меню займає, зазвичай, цілу шпальту сайту, що звужує решту сайту, а значить і місця під зміст буде набагато меншою. Тому для сайту ТОВ «Трансфрахт» було обрано горизонтальне меню (див. рис. 3.1).

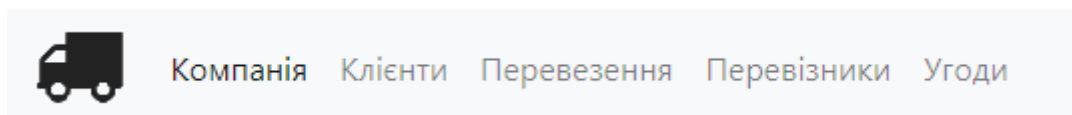


Рисунок 3.1 – Меню веб-сайту ТОВ «Трансфрахт»

Бізнес-стиль легко розпізнати за наявністю білих тонів, чітко розташованим блокам з інформацією, фіксованою шириною сайту, застосування банерної реклами та зображень, що відносяться до ділової тематики.

Шапка. (Header) так називається фрагмент контенту, що з'являється у верхній частині сайту. Це практично універсальний компонент, хоча в деяких ситуаціях без нього можна обійтися. Проте майже кожен сайт буде (або повинен) його мати.

Область констант. Тут, як правило, розташовуються відомості про основну торгову марку. Логотип найчастіше займає верхній лівий кут, але це не обов'язкове правило, а лише поширений стандарт, до якого звикла більшість користувачів. Тут же розташовується безліч елементів, про які відвідувачам сайту варто постійно пам'ятати.

Основна система навігації. З шапкою сайту зазвичай пов'язана основна система навігації. Саме вона дозволяє користувачам потрапити в глиб сайту.

Допоміжна система навігації. Сайти часто мають окрему вторинну систему навігації, в яку входять елементи, які не є ключовими для процесу продажів. Сюди часто потрапляють такі розділи, як «про компанію» і «контакти»

Пошук. Поле пошуку найчастіше розташовується в шапці (див. рис. 3.2). Це цілком логічно, адже саме тут знаходиться основна система навігації.



Рисунок 3.2 – Пошук тексту

Проектування баз даних, як правило, грає одну з ключових ролей в більшості проектів. Грамотно спроектована база дозволяє без особливих проблем вносити зміни, змінювати структуру системи.

Щоб використовувати базу даних, її потрібно звичайно ж створити.

Побудуємо схему бази даних та зв'язок її таблиць, що будуть створені (див. рис. 3.3).

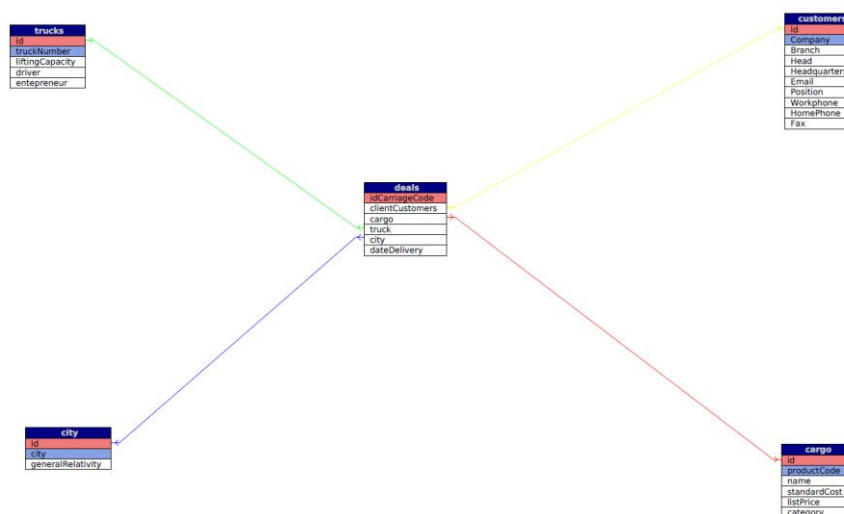


Рисунок 3.3 – Схема бази даних

В дану схему входять п'ять таблиць:

1) Таблиця cargo

Таблиця 3.1 – Таблиця реєстрації вантажу

id	int(11)	Унікальний ключ
productCode	text	Код вантажу
name	text	Назва вантажу
standartCost	float	Стандартна вартість
listPrice	float	Ціна за прејскурантом
category	text	Категорія

2) Таблиця city

Таблиця 3.2 – Таблиця реєстрації міст відправки вантажу

id	int(11)	Унікальний ключ
city	text	Назва міста
generalRelativity	text	Назва області

3) Таблиця trucks

Таблиця 3.3 – Таблиця реєстрації перевізників

id	int(11)	Унікальний ключ
truckNumber	text	Номер вантажівки
liftingCapacity	int(11)	Вантажопідйомність
driver	text	Водій
entrepreneur	text	Назва компанії

4) Таблиця customers

Таблиця 3.4 – Таблиця реєстрації клієнтів

id	int(11)	Унікальний ключ
Company	text	Назва компанії
Branch	text	Галузь
Head	text	Керівник
Headquarters	text	Головний офіс
Email	text	Електронна пошта
Position	text	Посада контактної особи
Workphone	text	Робочий телефон
HomePhone	text	Власний телефон
Fax	text	Факс

5) Таблиця deals

Таблиця 3.5 – Таблиця реєстрації угод

idCarriageCode	int(11)	Унікальний ключ
clientCustomers	int(11)	Код клієнта
cargo	int(11)	Код вантажу
truck	int(11)	Код вантажівки
city	int(5)	Код міста
dateDelivery	date	Дата угоди

Всі дані, окрім дати, формуються з попередніх таблиць.

Проектування баз даних, як правило, грає одну з ключових ролей в більшості проектів. Грамотно спроектована база дозволяє без особливих проблем вносити зміни, змінювати структуру системи.

Нормалізація даних - одне з найважливіших понять і концепцій реляційної системи. Нормалізована система зводить до мінімуму кількість надлишкових даних, при цьому зберігаючи їх цілісність. Нормалізованою можна назвати базу

даних, в якій всі таблиці дотримуються правил нормальних форм. Нормальна форма - набір правил, які показують, як треба організувати дані, що б вони були нормалізованими. З моменту заснування реляційних систем з'явилося безліч нормальних форм, але важливо розуміти, що використання кожної нової форми підвищує навантаження на систему. Тому, в більш складних системах завжди потрібно вміти знайти компроміс між ступенем нормалізації і продуктивності.

Отже, ми розглянули один з важливих етапів проектування бази даних.

3.2. Результати реалізації програми

До складу проекту входять наступні файли: *.html, *.css, *.php, *.sql, *.jpeg, *.png.

Усі зображення знаходяться в папці / image.

Файл CSS добре коментується та простий у використанні та налаштовується під власні потреби.

Файл, що містить базу даних:

1) tovtransfer.sql

Створення і просування сучасного інтернет-сайту являє собою творчий процес, що складається з ряду послідовних кроків. У цій справі важливий кожен інструмент і метод розкрутки сайту, адже від них залежить майбутнє нового проекту. Особливу цінність для будь-якого інтернет-ресурсу представляє інформаційне наповнення сайту оригінальним контентом, без якого інтернет-ресурс втрачає сенс і корисність для відвідувачів. Найчастіше, під цим поширеним терміном розуміють інформаційне наповнення веб-сайту.

Наведемо загальний вигляд сторінки меню «Компанія», що є головною сторінкою сайту у веб-браузері (див.рис. 3.4).

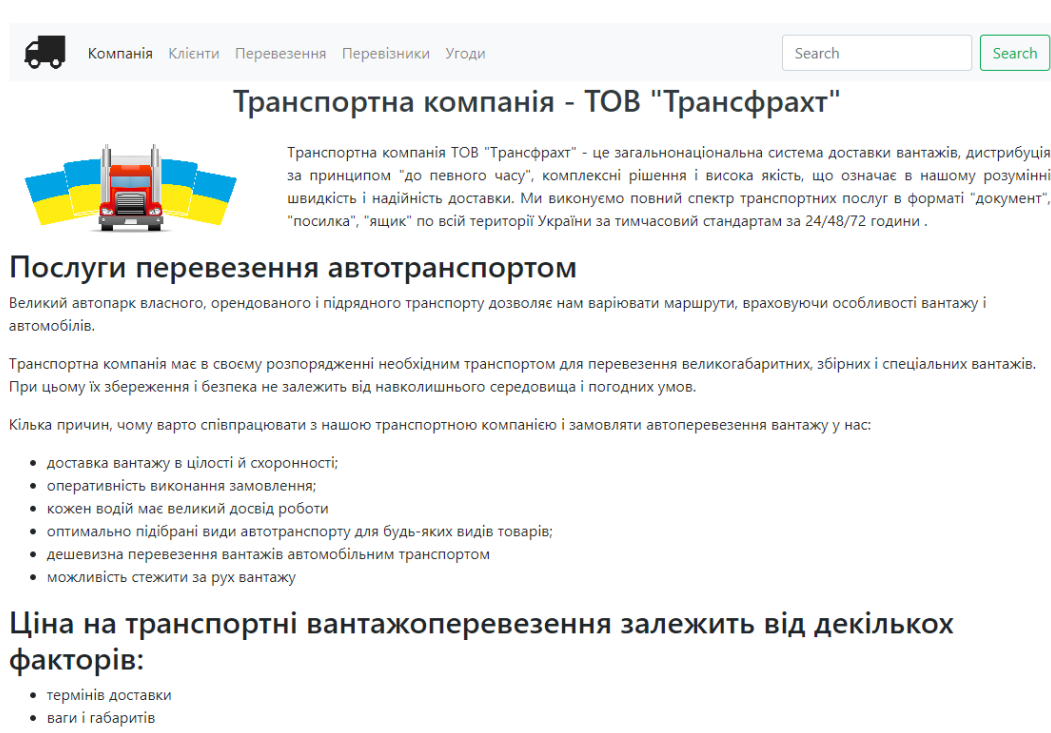


Рисунок 3.4 – Сторінка меню «Компанія»

Почнемо з основних вимог, яким повинен відповідати якісний контент:

Інформативність - на сайті повинна бути розміщена саме та інформація, яка цікавить цільових відвідувачів, тобто людей в розрахунку на яких і створювався даний сайт. Тому інформація повинна бути потрібною, актуальною, правдивою і корисною.

Грамотність - текст повинен бути грамотним в орфографічному, стилістичному, синтаксичному та лексичному відношенні, так само як і в технічному плані.

Цікавість - статті повинні бути цікаві користувачам, містити яскраві приклади, цікаві факти, що мають пряме відношення до теми.

Читабельність - текст повинен легко сприйматися, бути розділеним підзаголовками, виділеннями, нумерованими і маркованими списками та ін. «Якорями» утримують увагу читача на сторінці.

Унікальність - неодмінна умова розміщення матеріалу. Це необхідно і для просування в пошукових системах, і для відсутності претензій з точки зору захисту авторських прав.

Для створення наступних сторінок сайту використовуємо мову РНР.

В другому пункті меню «Клієнти» відображається основна інформація про клієнтів компанії ТОВ «Трансфракт» (див.рис.3.5).

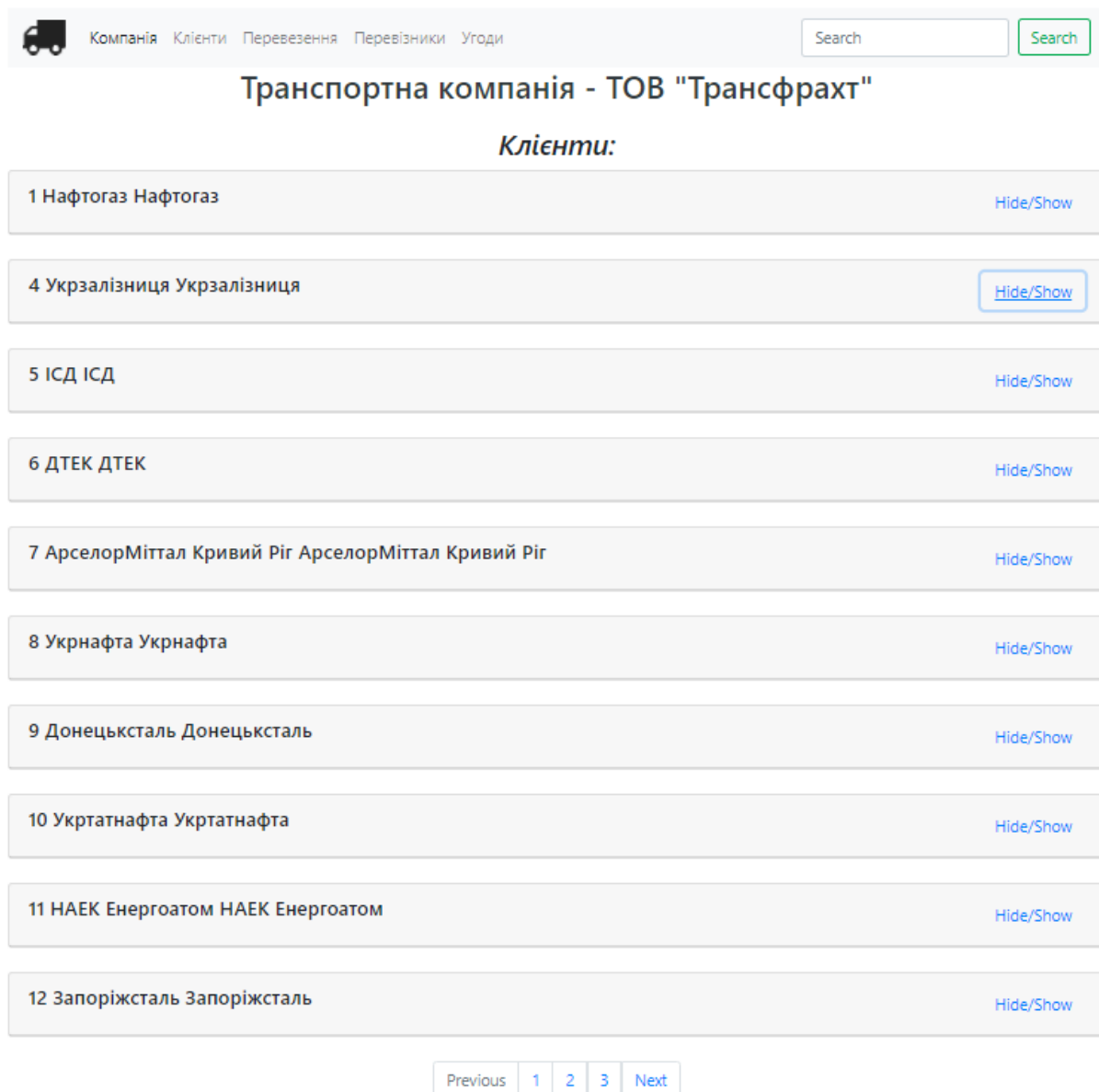


Рисунок 3.5 – Сторінка меню «Клієнти»

Всі клієнти представлені в згорнутому вигляді (перегляд назв компаній), при натисканні на кнопку (див.рис. 3.6) відкривається повний вигляд відгука (див.рис. 3.7).

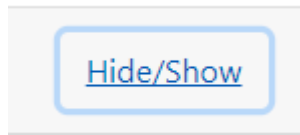


Рисунок 3.6 – Кнопка згортання та розгортання інформації

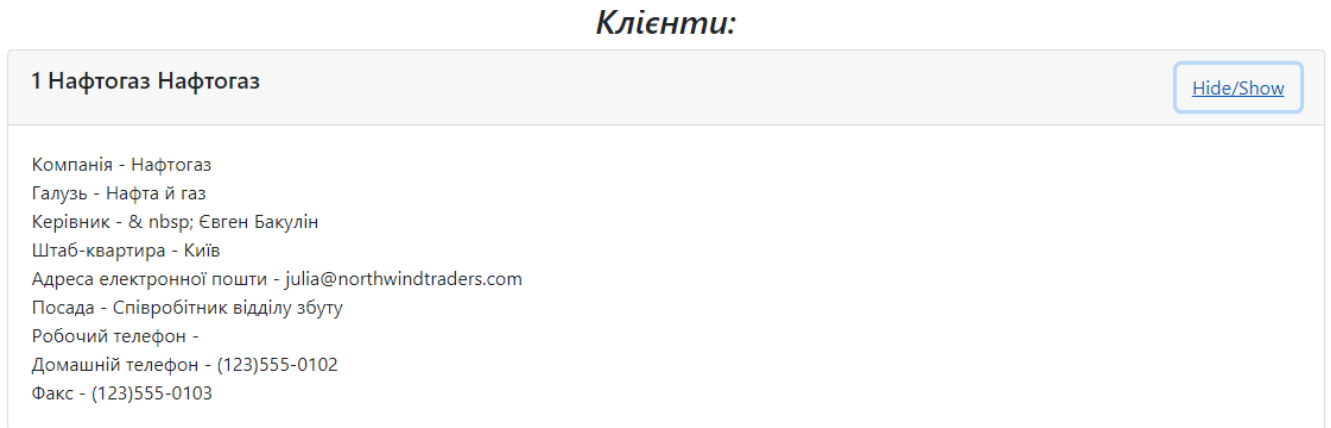


Рисунок 3.7 – Клієнти в розгорнутому вигляді

Клієнти у кількості 10 шт. на сторінці. Для гортання сторінок створюємо навігатор по сторінкам (див.рис. 3.8).

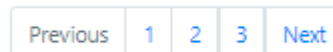


Рисунок 3.8 – Навігатор по сторінкам

В третьому пункті меню «Перевезення» відображається вибіркова інформація про перевезення (див.рис. 3.9).

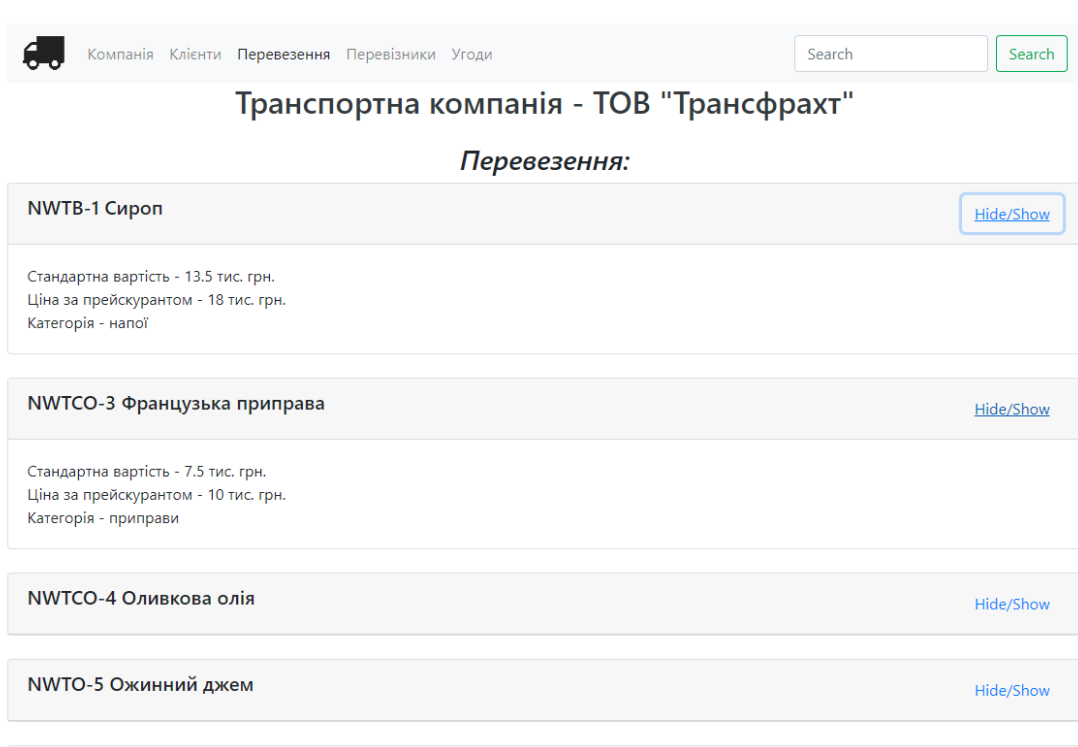


Рисунок 3.9 – Сторінка меню «Перевезення»

В четвертому пункті меню «Перевізники» відображаються дані вантажівок (див.рис. 3.10).

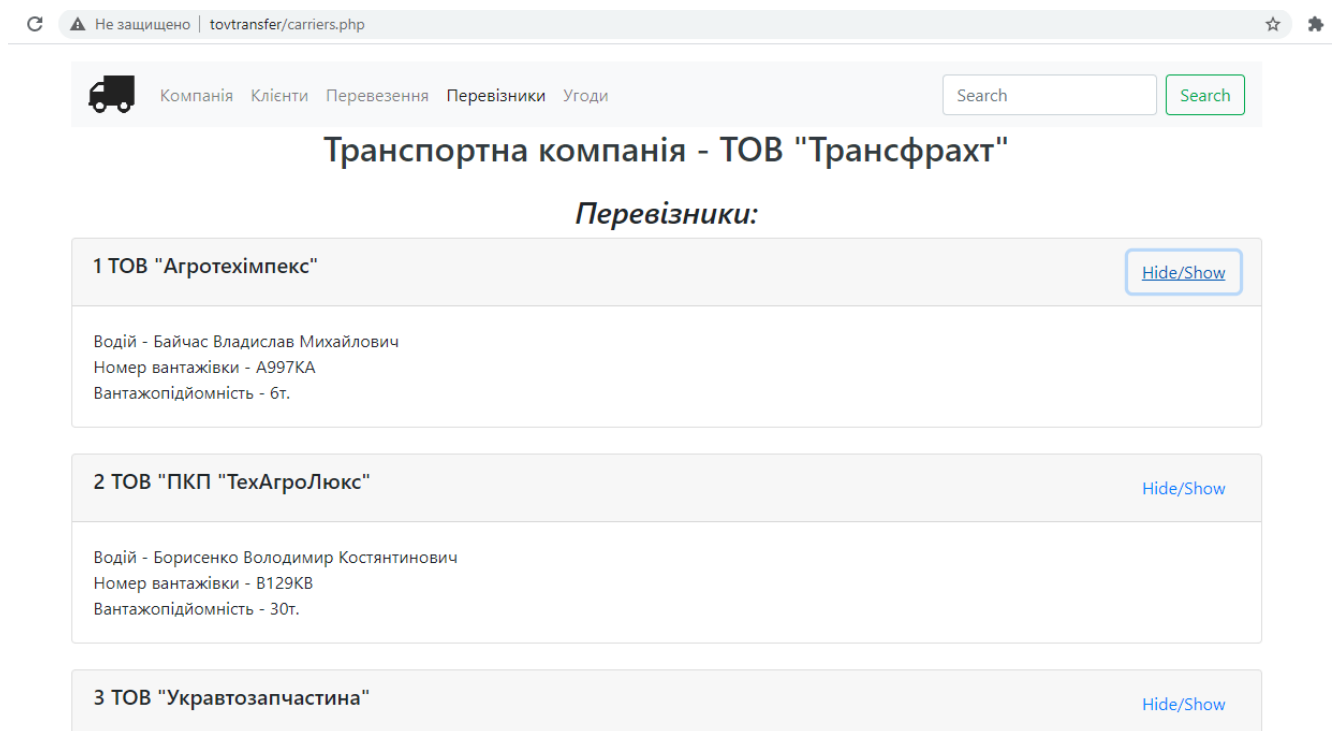


Рисунок 3.10 – Сторінка меню «Перевізники»

В п'ятому пункт меню «Угоди» відображаються дані угод, а саме: дата, клієнт, перевізник (див.рис. 3.11).

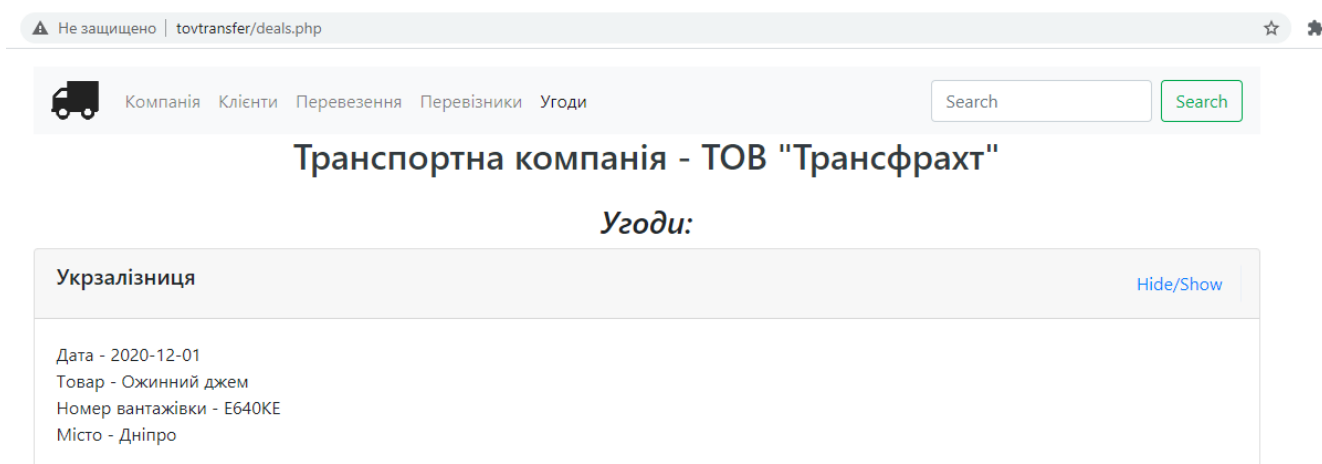


Рисунок 3.11 – Сторінка меню «Угоди»

Після створення логічного дизайну нам потрібно створити нашу базу даних відповідно до визначень, які ми створили. Для реалізації з реляційною СУБД це, ймовірно, передбачає використання SQL для створення таблиць та обмежень, які задовольняють опис логічної схеми, та вибір відповідної схеми зберігання (якщо СУБД дозволяє такий рівень управління).

Реалізація передбачає побудову бази даних відповідно до специфікації логічної схеми. Це включатиме специфікацію відповідної схеми зберігання, забезпечення безпеки, зовнішню схему тощо. На реалізацію сильно впливає вибір доступних СУБД, інструментів баз даних та операційного середовища. Існують додаткові завдання, окрім простого створення схеми бази даних та впровадження обмежень - дані повинні вноситися в таблиці, вирішуватись питання, що стосуються користувачів та користувацьких процесів, а управлінська діяльність, пов'язана з більш широкими аспектами корпоративного управління даними, повинна бути підтриманою.

Для цього була використана програма RHPMyAdmin. MySQL має тривірневу структуру: база даних; таблиці; записи.

Створимо базу даних «tovtransfer» за допомогою програми RHPMyAdmin,

після чого створимо в ній п'ять таблиць, які будуть містити інформацію з сайту, що розроблюється. У рядку «Створити нову таблицю в БД tovtransfer» надаємо їй назву «cargo» та вкажемо шість полів таблиці (див.рис. 3.12).

The screenshot shows the phpMyAdmin interface for the 'tovtransfer' database. The 'cargo' table is selected, and its structure and data are displayed. The table has 6 columns: id, productCode, name, standardCost, listPrice, and category. The data is as follows:

id	productCode	name	standardCost	listPrice	category
1	NWTB-1	Сироп	13.5	18	напої
2	NWTCO-3	Французька приправа	7.5	10	приправи
3	NWTCO-4	Оливкова олія	16.5	22	приправи
4	NWTO-5	Ожинний джем	16.01	21.35	Олія
5	NWTJP-6	Сушені груші	18.75	25	Варення і джеми
6	NWTDN-7	Каррі	22.5	30	Сухофрукти і горіхи
7	NWTS-8	Грецькі горіхи	30	40	соуси
8	NWTDN-14	Фруктовий салат	17.44	23.25	Сухофрукти і горіхи
9	NWTCFV-17	Шоколадні бісквіти	29.25	39	Фруктові та овочеві консерви
10	NWTBGM-19	Мармелад	6.9	9.2	Хлібобулочні вироби
11	NWTJP-6	Лаваш	60.75	81	Варення і джеми
12	NWTBGM-21	Пиво	7.5	10	Хлібобулочні вироби
13	NWTB-34	Тихоокеанські краби	10.5	14	напої
14	NWTCM-40	Атлантичні мідії	13.8	18.4	М'ясні консерви

Рисунок 3.12 – Таблиця «cargo»

Аналогічним чином створюємо ще чотири таблиці (див.рис. 3.13-3.16).

The screenshot shows the phpMyAdmin interface for the 'tovtransfer' database. The 'city' table is selected, and its structure and data are displayed. The table has 3 columns: id, city, and generalRelativity. The data is as follows:

id	city	generalRelativity
1	Київ	Київська
2	Харків	Харківська
3	Одеса	Одеська
4	Дніпро	Дніпровська
5	Донецьк	-
6	Запоріжжя	Запорізька
7	Львів	Львівська
8	Кривий Ріг	Криворізька
9	Миколаїв	Миколаївська
10	Севастополь	-

Рисунок 3.13 – Таблиця «city»

phpMyAdmin

Сервер: 127.0.0.1:3306 » База данных: tovtransfer » Таблица: trucks

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Отображение строк 0 - 24 (25 всего, Запрос занял 0,0003 сек.)

SELECT * FROM `trucks`

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

Показать все | Количество строк: 50 | Фильтровать строки: Поиск в таблице | Сортировать по ключу: Ниодного

+ Параметры

	id	truckNumber	liftingCapacity	driver	entrepreneur
<input type="checkbox"/>	1	A997KA	6	Байчас Владислав Михайлович	ТОВ "Агротехімпекс"
<input type="checkbox"/>	2	B129KB	30	Борисенко Володимир Костянтинович	ТОВ "ПКП "ТехАгроЛюкс"
<input type="checkbox"/>	3	C702KC	27	Верес Олег Олександрович	ТОВ "Укравтозапчастина"
<input type="checkbox"/>	4	E640KE	27	Годунок Ярослав Миколайович	ООО "СП Стан-Комплект"
<input type="checkbox"/>	5	H255KH	20	Грона Євгеній Миколайович	ТОВ "СТАНЭКСИМ- Україна"
<input type="checkbox"/>	6	I273KI	25	Грощенко Олександр Володимирович	ТОВ "Амкодор-Укрспецмаш"
<input type="checkbox"/>	7	M897KM	34	Денисов Олександр Анатолійович	ТОВ з/І «Торговий Дом «МАЗ-Україна»
<input type="checkbox"/>	8	O336KO	10	Молодий Віктор Леонідович	ТОВ «Авто-МАЗ»
<input type="checkbox"/>	9	P738KP	31	Москвічов Олексій Миколайович	ЧП ПКП «Альфатекс»
<input type="checkbox"/>	10	T359KT	11	Петровскіс Ігор Євалдсович	ТОВ «Київський Республіканський Автоцентр»
<input type="checkbox"/>	11	X539KX	35	Пилипчук Владислав Володимирович	ТОВ «Сучасні вантажівки»
<input type="checkbox"/>	12	A441HA	18	Полянція Андрій Вікторович	ТОВ «УкрЄвроМаз»
<input type="checkbox"/>	13	B326HB	11	Сидоренко Володимир Григорович	ТОВ «МазТрансСервіс»
<input type="checkbox"/>	14	C734HC	4	Тищенко Богдан Васильович	ТОВ "Агроресурс"
<input type="checkbox"/>	15	E839HE	6	Федорчук Анатолій Соловійович	ТОВ «Прайм Україна»
<input type="checkbox"/>	16	H915HN	30	Федорчук Артем Ігорович	ТОВ «МакТехБуд»
<input type="checkbox"/>	17	I815NI	29	Федорчук Олександр Миколайович	ТОВ «Агротехніка»
<input type="checkbox"/>	18	K762HK	24	Царюк Євген Володимирович	ТОВ "УкрАвтоКомплект ЛТД"

Консоль

Рисунок 3.14 – Таблица «trucks»

phpMyAdmin

Сервер: 127.0.0.1:3306 » База данных: tovtransfer » Таблица: customers

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Отображение строк 0 - 49 (50 всего, Запрос занял 0,0003 сек.) [id: 1... - 52...]

SELECT * FROM `customers` ORDER BY `id` ASC

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

1 > >> | Показать все | Количество строк: 50 | Фильтровать строки: Поиск в таблице | Сортировать по ключу: id (ASC)

+ Параметры

	id	Company	Branch	Head	Headquarters	Email	Position	Workphone	HomePhone
<input type="checkbox"/>	1	Нафтогаз	Нафта й газ	Євген Бакулін	Київ	julia@northwindtraders.com	Співробітник відділу збуту	(123) 555-0100	(123) 555-0102
<input type="checkbox"/>	4	Укрзалізниця	Транспорт	Володимир Козак	Київ	maria@northwindtraders.com	Співробітник відділу збуту	(123) 555-0100	(123) 555-0102
<input type="checkbox"/>	5	ІСД	Металургія	Олег Мкртчян	Донецьк	nikolay@northwindtraders.com	Начальник відділу збуту	(123) 555-0100	(123) 555-0102
<input type="checkbox"/>	6	ДТЕК	Електроенергетика	Максим Тимченко	Київ	vadim@northwindtraders.com	Співробітник відділу збуту	(123) 555-0100	(123) 555-0102
<input type="checkbox"/>	7	АрселорМіттал Кривий Ріг	Металургія	Артем Поляків	Кривий Ріг	sergey@northwindtraders.com	Співробітник відділу збуту	(123) 555-0100	(123) 555-0102
<input type="checkbox"/>	8	Укрнафта	Нафта	Пітер Ванхеке	Київ	inna@northwindtraders.com	координатор продаж	(123) 555-0100	(123) 555-0102
<input type="checkbox"/>	9	Лонейксталь	Металургія	Олександр	Донецьк	darya@northwindtraders.com	Співробітник	(123) 555-	(123) 555-

Консоль

127.0.0.1/openserver/phpmyadmin/sql.php?server=1&db=tovtransfer&table=customers&pos=0

Рисунок 3.15 – Таблица «customers»

Сервер: 127.0.0.1:3306 » База данных: tovtransfer » Таблица: deals

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии

Отображение строк 0 - 11 (12 всего, Запрос занял 0,0002 сек.) [idCarriageCode: 1... - 12...]

```
SELECT * FROM `deals` ORDER BY `idCarriageCode` ASC
```

☐ Профилирование [Построчное редактирование] [Исчистить]

☐ Показать все | Количество строк: 50 | Фильтровать строки: Поиск в таблице | Сортировка

+ Параметры

	idCarriageCode	clientCustomers	cargo	truck	city	dateDelivery
<input type="checkbox"/>	1	4	1	2	1	2020-12-01
<input type="checkbox"/>	2	5	1	2	1	2020-12-01
<input type="checkbox"/>	3	6	13	3	8	2020-12-02
<input type="checkbox"/>	4	7	13	3	8	2020-12-02
<input type="checkbox"/>	5	8	1	17	1	2020-12-03
<input type="checkbox"/>	6	9	1	17	1	2020-12-03
<input type="checkbox"/>	7	4	9	24	27	2020-12-04
<input type="checkbox"/>	8	5	9	24	27	2020-12-04
<input type="checkbox"/>	9	9	42	18	4	2020-12-02
<input type="checkbox"/>	10	8	42	18	4	2020-12-02
<input type="checkbox"/>	11	71	43	19	9	2020-11-09
<input type="checkbox"/>	12	71	43	19	9	2020-11-09

☐ Отметить все | С отмеченными: [иконки]

☐ Показать все | Количество строк: 50 | Фильтровать строки: Поиск в таблице | Сортировка

Рисунок 3.16 – Таблица «deals»

В результаті отримаємо базу даних, що складається з п'яти таблиць (див.рис. 3.17).

Сервер: 127.0.0.1:3306 » База данных: tovtransfer

Структура SQL Поиск Запрос по шаблону Экспорт Импорт Операции Привилегии Прогноз

Фильтры

Содержит слово:

Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
<input type="checkbox"/> cargo	[иконки]	45	InnoDB	utf8mb4_unicode_ci	32.0 КиБ	-
<input type="checkbox"/> city	[иконки]	43	InnoDB	utf8mb4_unicode_ci	32.0 КиБ	-
<input type="checkbox"/> deals	[иконки]	12	InnoDB	utf8mb4_unicode_ci	96.0 КиБ	-
<input type="checkbox"/> trucks	[иконки]	25	InnoDB	utf8mb4_unicode_ci	32.0 КиБ	-
<input type="checkbox"/> customers	[иконки]	69	InnoDB	utf8mb4_unicode_ci	128.0 КиБ	-
5 таблиц	Всего	194	InnoDB	utf8mb4_unicode_ci	320.0 КиБ	0 байт

☐ Отметить все | С отмеченными: [выпадающий список]

Печать Словарь данных

Создать таблицу

Имя: Количество столбцов: 4

Рисунок 3.17 – БД «tovtransfer»

Побудуємо зв'язки бази даних «tovtransfer» (див.рис. 3.18).

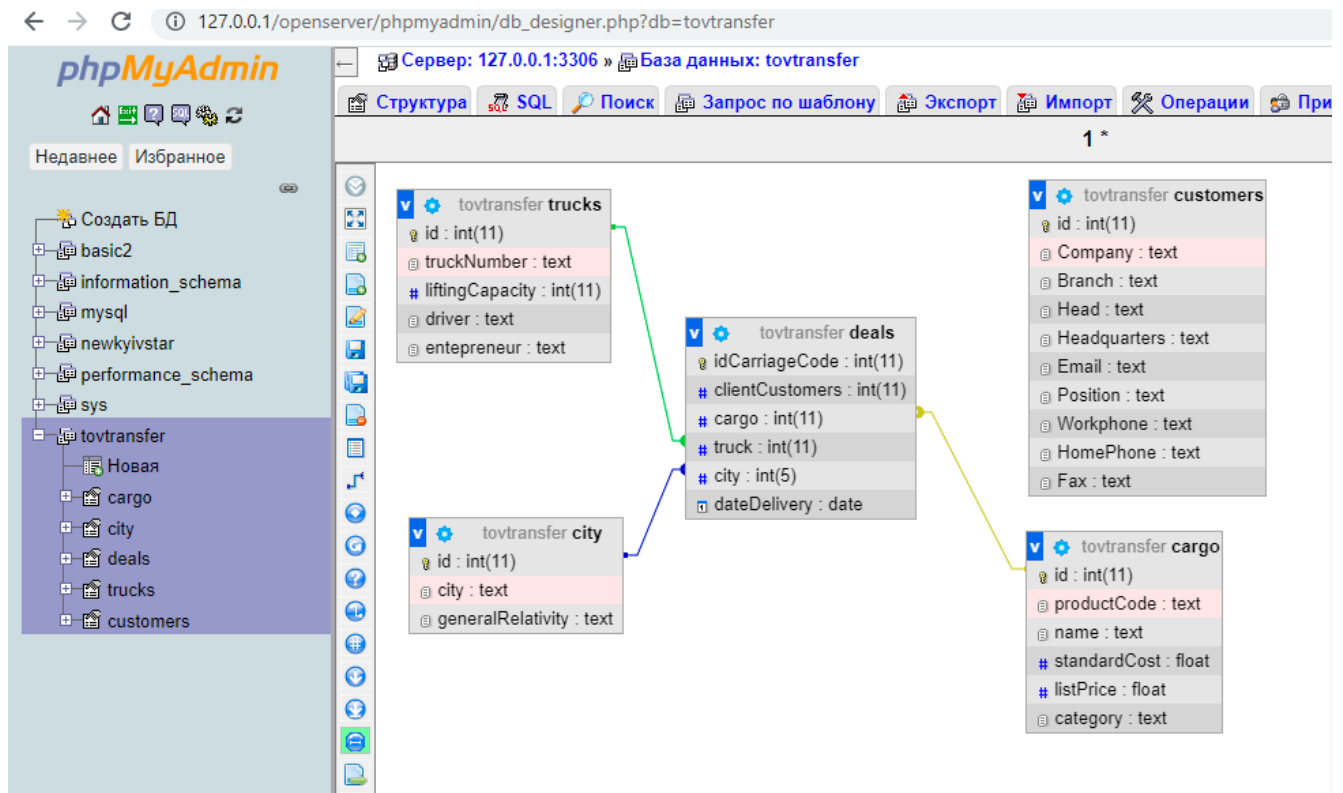


Рисунок 3.18 – Зв'язки БД «tovtransfer»

Тестування грає життєво важливу роль в процесі розробки і створення якісного програмного забезпечення. Необхідно серйозно ставитися до аналізу і проектування структурованого процесу, який забезпечить своєчасний і успішний випуск проекту.

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

Були проведені наступні заходи:

- 1) комплексний контроль якості;
- 2) підготовка тестової документації (плани тестування та ін.);
- 3) виявлення та локалізація помилок у функціонуванні сайту;
- 4) фіксування і відстеження помилок у функціонуванні сайту;
- 5) перевірка відповідності документації програмного продукту стандартам і

реально реалізованим функціям;

- 6) участь в розробці та впровадженні системи якості;
- 7) автоматизація тестування;
- 8) оцінка продуктивності розроблювальних програмних засобів на різних програмно-апаратних платформах і їх специфічних змінах.

При тестуванні ніяких критичних відмінностей у зовнішньому вигляді сайту в різних браузерах і при різних розширеннях не було виявлено.

Підбір правильного контенту для наповнення сайту забезпечив виконання всіх функцій інтернет-ресурсу:

9) Інформативна функція - кожен вид контенту (текстовий, графічний, відео- і аудіоматеріали) доносить до відвідувача корисну і достовірну інформацію.

10) Зручність користування (юзабіліті) - всі види контенту доступні і зручні для користувачів з будь-яким рівнем підготовки.

11) Навігаційна функція - інтуїтивно-зрозуміле і логічне розміщення всіх елементів забезпечує легке і безпроблемне переміщення по сторінках сайту і швидкий пошук потрібної інформації.

12) Естетична функція - оригінальні статті і тексти для сайту лежать в основі дизайну кожного інтернет-ресурсу і є його візитною карткою.

ВИСНОВКИ

Метою даної роботи була розробка сайту ТОВ «Трансфрахт».

Проектування бази даних - це пошук способів задоволення функціональних вимог засобами наявної комп'ютерної технології з урахуванням заданих обмежень.

Проект даної роботи зі створення бази даних включав в себе наступні етапи: визначення аналіз вимог до бази даних, стратегії побудови системи, проектування бази даних, тестування, реалізація, і впровадження бази даних.

Проектування починається з моменту прийняття стратегічних рішень і триває на етапах реалізації та тестування.

- 1) Процес проектування бази даних охоплює основні сфери.
- 2) Проектування об'єктів бази даних для представлення даних предметної області в базі даних.
- 3) Проектування інтерфейсу взаємодії з базою даних, тобто проектування сайту.

Широко відомі методи проектування баз даних з'явилися в процесі розробки все більш складних Інформаційних Систем, які повинні були розглядати потреби не одного користувача, але й великих груп і колективів. Одна така інтегрована БД створювалася для вирішення багатьох завдань, кожна з яких використовувала тільки "свою" частину даних, що зазвичай, перетинається з частинами, використовуваними в інших завданнях. Тому найголовнішими методами проектування стали методи виключення надмірності в даних. Ці методи зв'язувалися з іншими засобами забезпечення логічної цілісності даних.

Було сформульовано принципову вимогу відділення програм від інтегрованих даних. Цей принцип спрямований на відчуження даних як ресурсу підприємства, важливий також тим, що консервативні за характером дані відділялися від прикладних програм, які могли часто зазнавати змін.

Іншою важливою проблемою проектування БД стало забезпечення

потрібних експлуатаційних параметрів, таких як обсяг зовнішньої пам'яті або час виконання різних операцій.

Створення корпоративних БД в умовах Нового Системного Проектування - діяльність, яка використовує багато методів класичного проектування, але вимагає іншої організації і багатьох додаткових та нових методів.

Після проведення аналізу було вирішено розробити веб-ресурс за допомогою веб-технологій HTML 5 – в якості базової розмітки та PHP – як основної мови програмування.

В якості середовища розробки було обрано Laravel, оскільки він швидкий і простий у використанні а також підтримує PHP і можливість створення баз даних.

В процесі розробки сайту для ТОВ «Трансфрайт» та оформлення атестаційної роботи були закріплені теоретичні знання, застосовані навички об'єктно-орієнтованого програмування, здійснено оформлення документації в текстових та графічних редакторах.