

Національний університет «Полтавська політехніка імені Юрія Кондратюка»
(повне найменування вищого навчального закладу)

Навчально-науковий інститут інформаційних технологій та робототехніки
(повна назва інституту)

Кафедра комп'ютерних та інформаційних технологій і систем
(повна назва кафедри)

**Пояснювальна записка
до дипломного проекту (роботи)**

магістра

(рівень вищої освіти)

на тему

Інформаційна система “Тюнінг автомобілів з елементами інтелектуального
пошуку інформації”

Виконав: студент 2 курсу, групи 601-ТН
спеціальності

122 Комп'ютерні науки

(шифр і назва спеціальності)

Деменко Р. Ю.

(прізвище та ініціали)

Керівник Головко Г. В.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Полтава – 2021 року

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ « ПОЛТАВСЬКА
ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ**

**КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
І СИСТЕМ**

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

спеціальність 122 «Комп'ютерні науки»

на тему

**Інформаційна система “Тюнінг автомобілів з елементами
інтелектуального пошуку інформації”**

Студента групи 601-ТН Деменка Ростислава Юрійовича

Керівник роботи
кандидат технічних наук,
доцент Головка Г. В.

Завідуючий кафедри
кандидат технічних наук,
доцент Головка Г. В.

РЕФЕРАТ

Кваліфікаційна робота магістра: 62 с., 34 малюнки, 1 додаток, 23 джерела.

Об'єкт дослідження: інформаційні системи, що стосуються тюнінгу автомобілів усіх міст України. Дізнатися про діяльність цих інформаційних систем.

Предмет досліджень: інструментальні засоби розробки інформаційної системи.

Мета кваліфікаційної роботи: розроблення і впровадження інформаційної системи тюнінгу автомобілів в Україні.

Ключові слова: web-сайт, база даних, система управління базами даних, персональний комп'ютер, web-сервер, панель управління.

ABSTRACT

The explanatory note contains: 62 pages, 34 Figures, 23 Sources, 1 appendix.

Object of research: information systems related to car tuning in all cities of Ukraine. Learn about the activities of these information systems.

Subject of research: tools for developing an information system.

The purpose of the qualification work: development and implementation of an information system for car tuning in Ukraine.

Keywords: website, database, database management system, personal computer, web server, control panel.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	6
ВСТУП	7
РОЗДІЛ 1 ІНФОРМАЦІЙНИЙ ОГЛЯД СИСТЕМИ	8
1.1 Постановка завдання.....	8
1.2 Поняття та види інформаційних систем	8
1.3 Етапи розвитку інформаційних систем.....	9
1.4 Візуальні класифікації	10
1.5 Види інформаційних систем	11
1.6 Призначення розробки.....	14
1.7 Вимоги до складу виконуваних розділів	15
1.8 Інтелектуальний пошук інформації.....	17
1.9 Огляд існуючих в Інтернеті інформаційних систем про надання послуг тюнінгу автомобілів в містах України.....	17
1.10 Постановка задачі дослідження	20
РОЗДІЛ 2 ПРОЕКТНІ РІШЕННЯ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	22
2.1 Діяльність «CarRepairServices».....	22
2.2 Розробка програмного забезпечення.....	24
2.3 Контекстна діаграма.....	24
2.4 Діаграма декомпозиції	25
2.5 Діаграма прецедентів	26
2.6 Діаграма діяльності	28
2.7 Розробка ER–моделі предметної області	29

	5
2.8 Структура інформаційної системи	30
2.9 Вибір серверної мови програмування.....	32
2.10 UML діаграми	33
2.11 Вибір локального сервера.....	35
2.12 Вибір візуального середовища програмування.....	37
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ.....	39
3.1 Розробка структури	39
3.2 Вигляд зі сторони користувача	39
РОЗДІЛ 4 ТЕСТУВАННЯ	46
4.1 Тестування ІС.....	46
4.2 Тест-план	46
ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТОК А КОД ПРОГРАМИ	53

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ІТ – інформаційні технології.

КІС – комп’ютерна інформаційна система.

UML–уніфікована мова моделювання, використовується у парадигміоб’єктно-орієнтованого програмування. Є невід’ємною частиною уніфікованогопроцесу розробки програмного забезпечення.

Веб-ресурс (сайт)– сукупність веб-сторінок, доступних у мережі, які об’єднані як за змістом, так і навігаційно.

ІС (інформаційна система) –сукупність організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів.

Ієрархія– поділ на вищі й нижчі посади, чини; суворий порядок підлеглості нижчих щодо посади або чину осіб вищим.

Таксономія– механізм, що дозволяє створювати довільну кількість тематичних категорій для вмісту сайту і асоціювати їх з модулями, що забезпечують введення і виведення інформації.

ВСТУП

Інформаційні системи[1] використовуються як механізм спілкування між власниками сайту і його користувачами, а іноді – між самими користувачами. Власники сайтів зазвичай ставлять завдання і визначають основні правила взаємодії, в той час, як користувачі – це ті люди, що відвідують сайт і намагаються користуватися представленим на ньому вмістом або його можливостями. Канал зв'язку між власником сайту і його відвідувачем може змінюватися. Найчастіше власники сайтів надають користувачам інформацію для її споживання, роблячи з цього частково одностороння взаємодія.

Створення інформаційної системи на основі Web-технологій[2] забезпечує можливість віддаленого доступу до системи клієнтам і співробітникам компанії в будь-який момент часу. За допомогою Web-орієнтованої системи на основі сайту відбувається швидке проходження і обробка запитів. Впровадження інформаційної системи на основі web-порталу забезпечує більш ефективну роботу підприємства за рахунок автоматизації основних бізнес-процесів при взаємодії з клієнтами, а також просування рекламних послуг в мережі для залучення нових клієнтів.

Інтернет все більше входить в різні сфери економіки і бізнесу. Сайт компанії представляє недорогий спосіб просування своїх послуг більш широкому колу своїх споживачів. У зв'язку з цим для будь-якої компанії важливо мати своє веб-представництво, інтегроване в інформаційну систему компанії.

Метою даної роботи є створення індивідуального повнофункціональної системи для пошуку та замовлення запчастин для тюнінгу автомобілів.

РОЗДІЛ 1

ІНФОРМАЦІЙНИЙ ОГЛЯД СИСТЕМИ

1.1 Постановка завдання

Інформаційна система зараз важлива тема для будь-якої сфери обслуговування. Якщо слідкувати на новинами, то найбільш популярне зараз саме тюнінг автомобілів серед молоді та не тільки молоді. З кожним днем кількість бізнесменів, які зв'язують свій бізнес з глобальною павутиною, збільшується на сотні тисяч осіб.

Метою розробки данної інформаційної системи був попит на ринку серед автомобільної індустрії. Серед розглянутих систем, подібних моєї теми були занотовані помилки та уникнення на стадії розробки.

1.2 Поняття та види інформаційних систем

Під системою[5] мається на увазі об'єкт, який одночасно реалізований як одне ціле. Системи значно відрізняються між собою як по составу так і цілям.

Інформаційна система – це система, яка здійснюється за допомогою:

- отриманням вхідних даних;
- обробкою цих даних або їх заміною;
- виводом результату;

Простою інформаційною системою назвемо систему, елементи якої функціонують відповідно до правил, породжених одним і тим же протіворечивим безліччю аксіом.

Складною інформаційною системою назвемо систему, яка містить елементи, що функціонують відповідно до правил, породжених відмінними один від одного множинами аксіом. При цьому допускається, що серед

правил функціонування різних елементів можуть бути взаємопротирічливі правила і цілі.

Таблиця 1.1 –Поняття ІС

Система	Елементи системи	Головна мета системи
Фірма	Люди, обладнання, матеріали, будівлі та ін.	Виробництво товарів
Комп'ютер	Електронні та електромеханічні елементи, лінії зв'язку та ін.	Обробка даних
Телекомунікаційна система	Комп'ютери, модеми, кабелі, мережеве програмне забезпечення та ін.	Передача інформації
Інформаційний система	Комп'ютери, комп'ютерні мережі, люди, інформаційне та програмне забезпечення	Виробництво професійної інформації

1.3 Етапи розвитку інформаційних систем

Історичний розвиток ІС і цілі їх використання в різні часові інтервали представлені в Таблиці 1.2.

Таблиця 1.2 –Історичний розвиток ІС

Період часу	Концепція використання інформації	Вид інформаційних систем	Мета використання
1950-1960 рр.	Паперовий потік розрахункових документів	Інформаційні системи обробки розрахункових документів на електромеханічних бухгалтерських машинах	Прискорення швидкості перевірки документів. Полегшення процедури обробки рахунків і розрахунку зарплати
1960-1970 рр.	Базова набір для підготовки звітів	Управлінські ІС для виробничої інформації	Прискорення процесу підготовки звітності
1970-1980 рр.	Управлінський контроль реалізації (продажів)	Системи підтримки прийняття рішень. Системи для вищої ланки управління	Вибір найбільш раціонального рішення
1980-2000 рр.	Інформація-стратегічний ресурс	Стратегічні інформаційні системи. Автоматизовані офіси	Виживання і процвітання фірми

Перші інформаційні системи з'явилися в 50-х рр. у ці роки вони були призначені для обробки рахунків і розрахунку зарплати, а реалізовувалися на електромеханічних бухгалтерських рахункових машинах. Це призводило до деякого скорочення витрат і часу на підготовку паперових документів.

В 60-х рр. змінюється ставлення до інформаційних систем. Інформація, отримана з них почала використовуватися для звітності.

1.4 Візуальні класифікації

Групуючи сайти за візуальною ознакою, ми стикаємося з діапазоном, на одному боці якого – сайти, які в більш значній мірі ґрунтуються на текстовому вмісті, а на іншій – ті, що більше фокусуються на графічному поданні або зображеннях. В Internet є чотири найбільш поширених дизайнерських види:

- тексто-орієнтовані сайти. Такі сайти проєктуються з упором на текстовий вміст. Вони швидко завантажуються, а їх дизайн злегка мінімалістичний;
- сайти в стилі GUI. На цих сайтах дотримуються деякі угоди, пов'язані з графічним призначенням для користувача інтерфейсом (GUI – graphical user interface), взяті з проєктування програмного забезпечення, наприклад вирівняні догори рядки меню, піктограми та спливаючі вікна. Серед GUI-орієнтованих сайтів є і прості GUI-компоненти, що додаються здебільшого в тексто-орієнтовані сайти, і повнофункціональні Web-додатки зі спеціальними елементами призначеного для інтерфейсу користувача;
- метафоричні сайти. Метафоричні сайти черпають ідеї з "реальному житті". Наприклад, в дизайні та навігації сайту, присвяченого автомобілям, можуть бути залучені приладова панель і кермове колесо. Метафорично оформлений сайт, як правило, надзвичайно яскравий візуально або інтерактивний. Це може вводити в оману одних користувачів і залучати інших;

- експериментальні сайти. Творці таких сайтів намагаються робити речі, трохи відступаючи від норм. Наступних експериментальному стилю оформлення, нерідко залучаються творчість, непередбачуваність, новаторство і навіть хаотичність;

Безумовно, в Internet ми знаходимо змішання форм і потенційно нову класифікацію сайтів. Наприклад, як можна класифікувати портал, який забезпечує багатство вмісту, варіантів навігації й навіть функцій, пов'язаних зі спільнотами, і все це на одній сторінці? Безперечно, цей дизайнерський стиль застосовується на величезній кількості сайтів. Ми спостерігаємо приховане підвищення впливу інших категорій дизайну – для цього досить поглянути на такі жанри Web-дизайну, як сайти електронної комерції, особливо чисті сайти типу "каталог і кошик", а також сайти персональних мережових журналів.

1.5 Види інформаційних систем

За ознакою структурованості завдань (рис. 1.3).

При створенні або класифікації ІС постійно з'являються недоліки, яких є формально – математичний та алгоритмічний опис задачі. Від ступеня формалізації багато в чому залежать ефективність роботи всієї системи, а також рівень автоматизації, на основі одержуваної інформації.

Чим точніше математичний опис завдання, тим вище можливості комп'ютерної обробки даних і тим менше ступінь участі людини в процесі її вирішення. Це і визначає ступінь автоматизації завдання.

Розрізняють три типи завдань, для яких створюються інформаційні системи: структуровані (формалізовані), неструктуровані (неформалізовані) і частково структуровані.

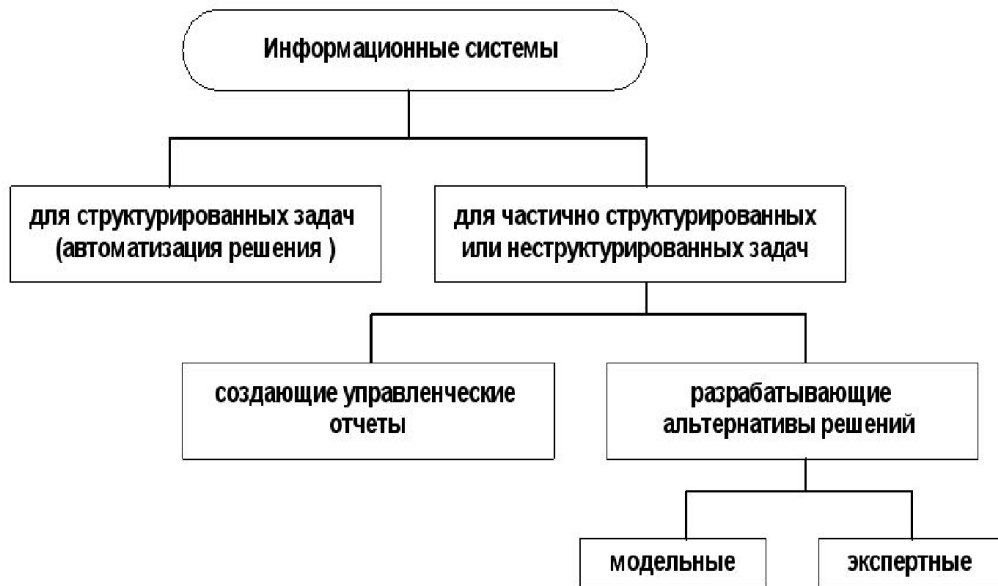


Рисунок 1.3 – Класифікація інформаційних систем за ознакою структурованості розв’язуваних задач

Структуроване (формалізована) завдання – це завдання, де мають ся елементи та взаємозв’язки між ними. Мета структурованого завдання полягає в висловленні основної ідеї у вигляді математичної математичної моделі, що являє собою максимально приближене рішення. Завдання такого типу досить складно вирішити одразу, тому доводиться повторювати цикл декілька разів, що затягує час на пошук рішення. Даний тип завдання використовують з метою вирішення структурованих задач де необхідна повна автоматизація їх вирішення, тобто зведення ролі людини до нуля.

Неструктуроване (неформалізоване) завдання – завдання, в якому неможливо виділити елементи і встановити між ними зв’язки. Рішення неструктурованих задач через неможливість створення математичного опису і розробки алгоритму пов’язано з великими труднощами. Можливості використання тут інформаційної системи невеликі. Рішення в таких випадках приймається людиною з евристичних міркувань на основі свого досвіду і, можливо, непрямой інформації з різних джерел.

Зауважимо, що в практиці роботи будь-якої організації існує порівняно небагато повністю структурованих або абсолютно неструктурованих завдань.

Про більшість завдань можна сказати, що відома лише частина їх елементів і зв'язків між ними. Такі завдання називаються частково структурованими. У цих умовах можна створити інформаційну систему. Одержувана в ній інформація аналізується людиною, який буде грати визначальну роль. Такі інформаційні системи є автоматизованими, так як в їх функціонуванні бере участь людина.



Рисунок 1.4 – Класифікація інформаційних систем за різними ознаками

За характером використання інформації.

Інформаційно-пошукові системи (див. рис. 1.4) виробляють ввід інформації, їх систематизацію та зберігання, вивід результат за запитом користувача без перетворень даних.

Інформаційно-вирішальні системи виконуються за певним алгоритмом який прописаний в кодї. Данні алгоритми можно подїлити на два класа – це керуючі та радники.

Керуючі ІС генерють данні, на основі яких користувач приймає рішення. Данні системи характеризуються за типом розрахункового характеру і обробкою даних значних масштабів. Зразком такої інформації можуть виступати система швидкого планування випуску продукції, система бухгалтерського обліку.

Радники ІС генерують інформацію, яка приймається людиною до відома і не перетворюється негайно в серію конкретних дій. Ці системи мають вищий рівень інтелекту для них характерна обробка знань, а не даних.

За сферою застосування

ІС організаційного застосування потрібні для автоматизації функцій. За наглядом, ІС використовуються як в промислових фірмах так і в промислових об'єктах: готелях, банках, тогових фірмах.

Основними функціями подібних ІС являється – оперативний контроль та регулювання, аналіз, бухгалтерський облік.

1.6 Призначення розробки

Корпоративні сайти створюються переважно для підтримки бізнесу якої-небудь організації. Основною аудиторією корпоративного сайту є потенційні та справжні клієнти організації. Додаткова аудиторія часто включає потенційних і дійсних інвесторів, службовців і зацікавлені сторони організації, такі як засоби інформації. З урахуванням такої змішаної аудиторії основні цілі корпоративних сайтів включають:

- поширення основної інформації – сайт використовується для розповсюдження інформації про продукцію та послуги, що надаються

організацією. інша надається інформація в основному включає дані про те, як зв'язатися з компанією способами, відмінними від Web;

- підтримку – частина сайту може бути призначена для надання клієнтам інформаційної допомоги щодо ефективного користування продуктами та послугами, наданих організацією;

- відносини з інвесторами – це відкриті товариства або компанії, які займаються пошуками інвесторів, вони можуть створити систему або частинусистеми для опублікування інформації про поточну фінансове положення в компанії, а також про потенціальні можливості для інвестування;

- зв'язок з громадськістю –переважна більшість компаній використовують ІС для надання інформації різним організаціям, які займаються парсингом даних, а також добровільного оприлюднення основної інформації про фірму;

- пошук службовців –ІСрегулярно використовують для публікування оголошень про пошук співробітників на роботу та переваги роботи в компанії.

1.7Вимоги до складу виконуваних розділів

Перед розробкою макету, Web-сайт необхідно розбити на кілька розділів.

Сторінка «Home» – відображає пошукове поле, послуги, відгуки, компанії з якими сайт співпрацює та контактні данні підтримки

Сторінка «Shop» – один з найбільш важливих розділів сайту – відображає список товарів, з можливістю сортування по категоріям, ціні та ін.

Сторінка «Contact us»– складається з форми, яку користувач заповнює для зв'язку з підтримкою сайту. Також на цій сторінці представлені контактні данні офісу компанії.

Сторінка «FAQ's» – містить у собі інструкцію з користування сайту
 Кнопка «Appointment»– перенаправляє роботодавця на форму, у котрій він може швидко та зручно записатися на обслуговування.

Також повинна бути розроблена можливість реєстрації нових користувачів. Підсумовуючи вище сказане, потрібно розробити авторизацію вже існуючих користувачів.

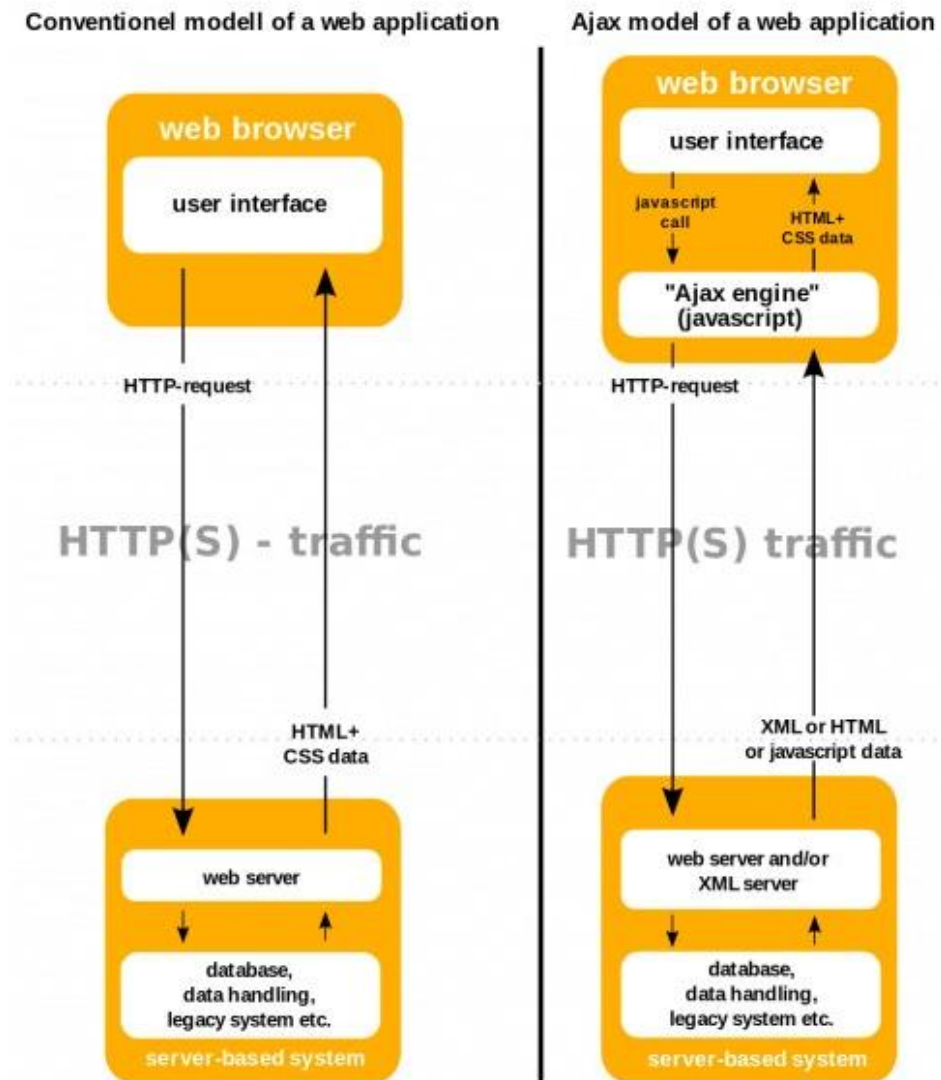


Рисунок 1.5–Схеми роботи пошуку

1.8 Інтелектуальний пошук інформації

Інтелектуальний пошук працює за аналогом пошукових систем. Іншими словами, інтелектуальний можна назвати розумний пошук. Уже на етапі вводу слів, він видаватиме інформацію в схожу на ту що введено. Схема роботи данного пошуку можете переглянути на рисунку 5. Далі система за спеціальним алгоритмом впорядковує знайдену інформацію і виводить назву, які оцінені як більш підходящі, в початок списку.

1.9 Огляд існуючих в Інтернеті інформаційних систем про надання послуг тюнінгу автомобілів в містах України

Зазвичай подібні сайти створюють з гарним дизайном, але чи достатньо цього користувачеві?! Крім того потрібно добре інформувати користувача, надати максимально інформації про послугу, аби в того не виникало зайвих питань. Сайт має бути доступним та зрозумілим для будь-якого клієнта. Повинна бути проста навігація, щоб кожен зміг замовити деталь без проблем.

Зараз розглянемо приклади деяких сайтів, що стосуються продажу запчастин для тюнінгу автомобілів:

1. «DesignTuning» Тюнінг автомобілів в місті Луцьк design-tuning.com представлений на рис. 1.6 та рис. 1.7

Недоліки представленого сайту:

- недоліків на сайті не знайдено;

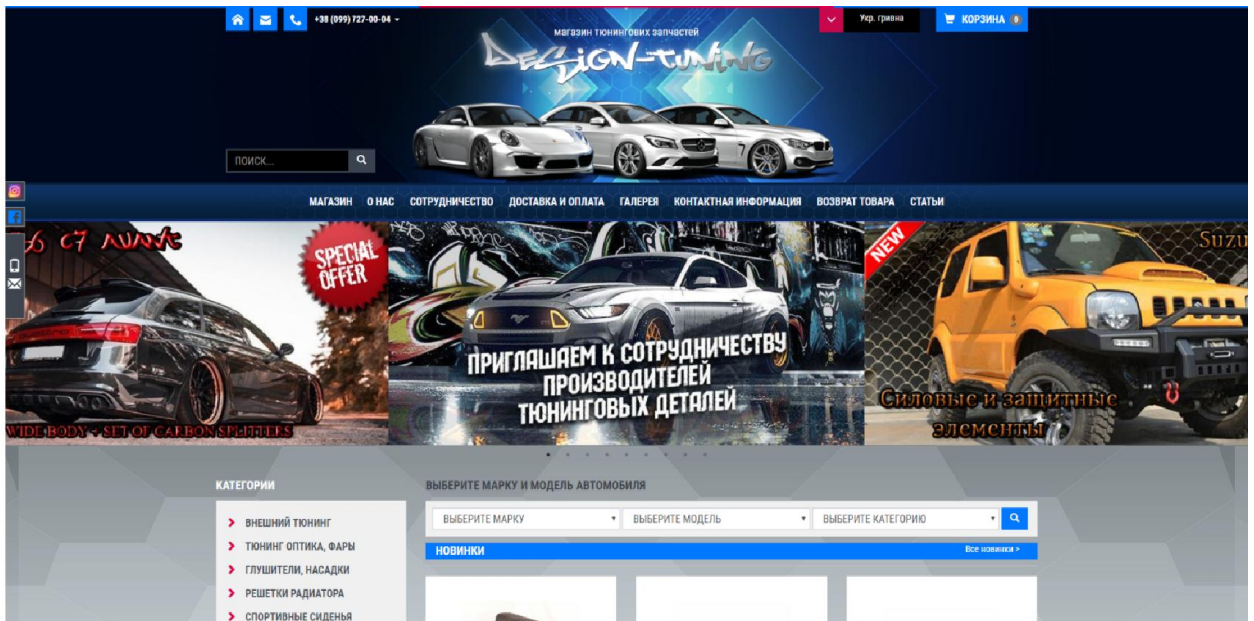


Рисунок 1.6 – Тюнінг автомобілів у Луцьку design-tuning.com

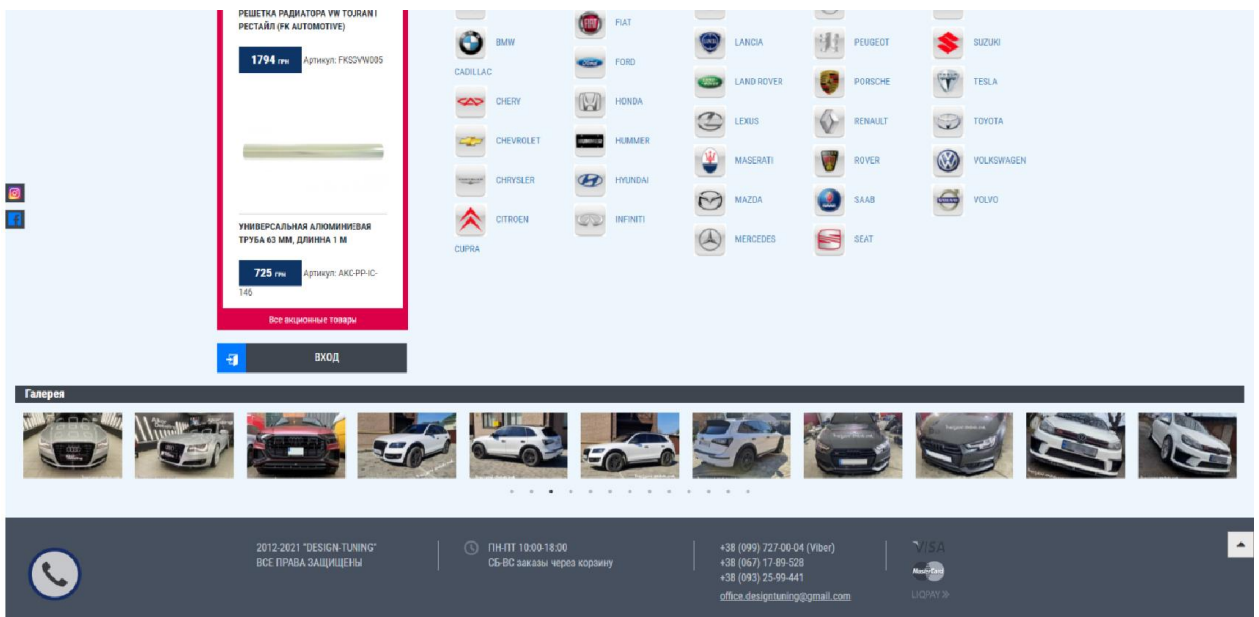


Рисунок 1.7 – Тюнінг автомобілів у Луцьку design-tuning.com

2. «SPORTCAR» Тюнінг автомобілів у Луцьку sport-car.com.ua представлений на рис. 1.8 та рис. 1.9

Недоліки представленого сайту:

- застарілий дизайн;

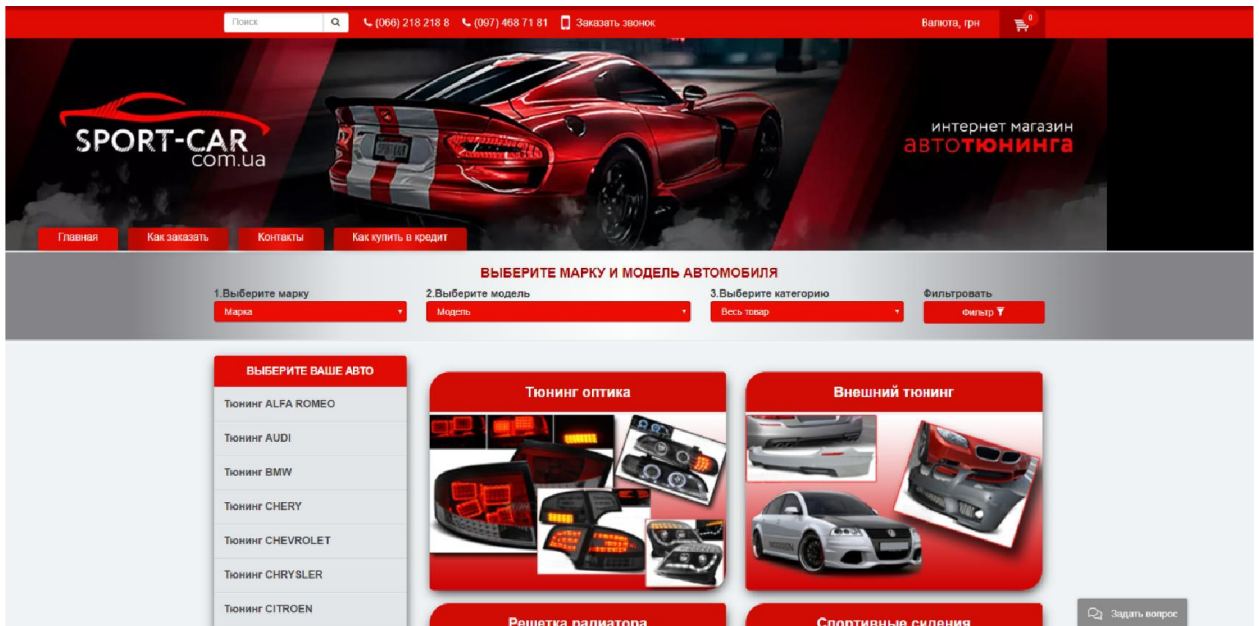


Рисунок 1.8 – Тюнінг автомобілів у Луцькуsport-car.com.ua

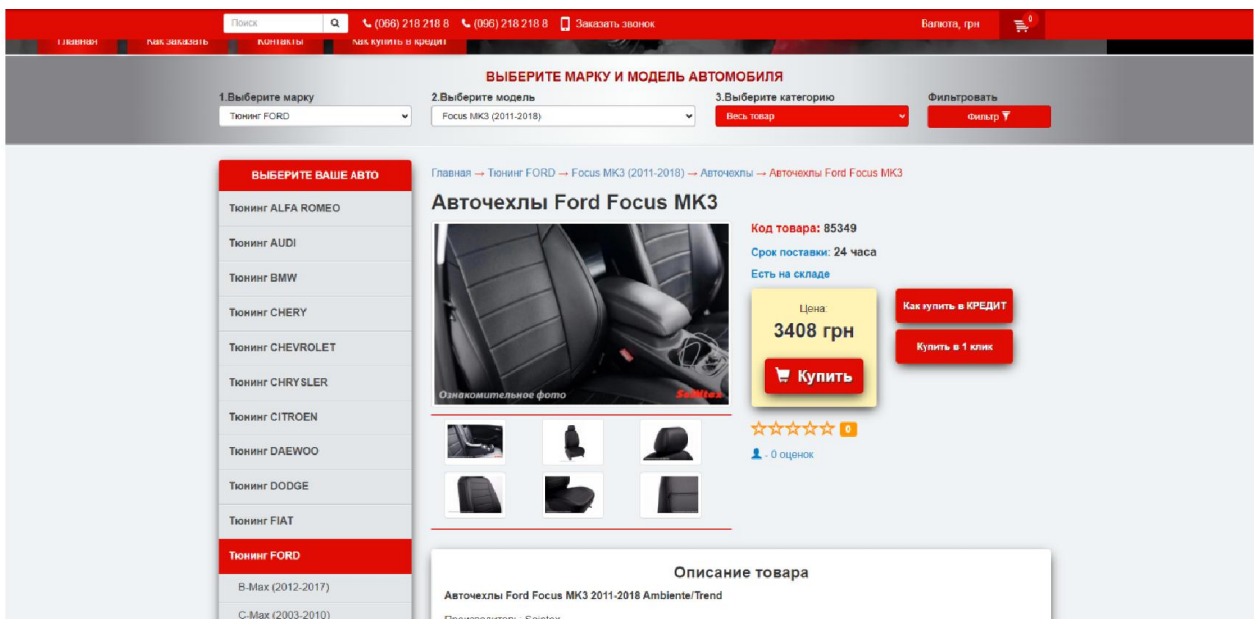


Рисунок 1.9 – Тюнінг автомобілів у Луцькуsport-car.com.ua

3. «AutoTuning» Тюнінгавтомобілів у Києві auto-tuning.in.ua представлений на рис. 1.10 та рис. 1.11

Недоліки представленої сайту:

- застарілий дизайн;

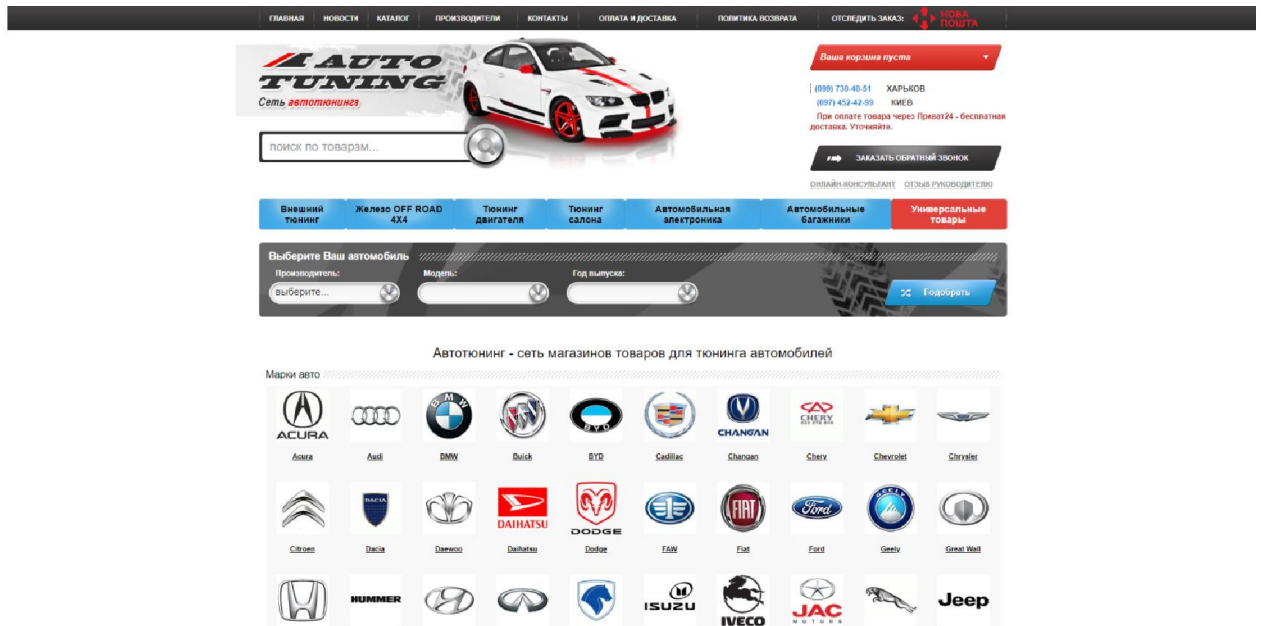


Рисунок 1.10 – Тюнинг автомобилей в мiстi Днiпро auto-tuning.in.ua

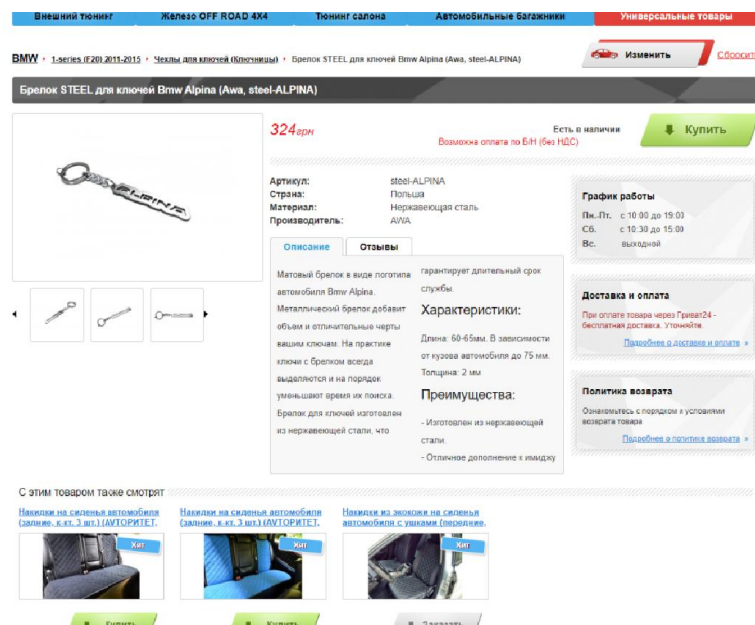


Рисунок 1.11 – Тюнинг автомобилей в мiстi Днiпро auto-tuning.in.ua

1.10 Постановка задачи дослідження

На основі проведеного аналізу наведених сайтів була поставлена задача дослідження: спроектувати та програмно реалізувати сайт, з надання послуг оренди автомобілів, який повинен забезпечувати виконання таких функцій:

- купівля товарів;

- можливість перегляду товару на авто;
- категорії;
- детальна інформація щодо товарів;
- можливість додавання нових товарів на сайт адміністрацією;
- повідомлення адміністрацією про діючі акції;
- зв'язок з довідковою службою, зручним для користувача, шляхом (телефон, електронна пошта, skype);

РОЗДІЛ 2

ПРОЕКТНІ РІШЕННЯСТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Діяльність «CarRepairServices»

Компанія «CarRepairServices» готова запропонувати Вам широкий спектр послуг в сфері тюнінгу вашого авто.

Ми пропонуємо замовникам повний спектр послуг – починаючи від косметичного ремонту і завершуючи повним переображенням вашого авто.



Рисунок 2.1 – Будівля «CarRepairServices»

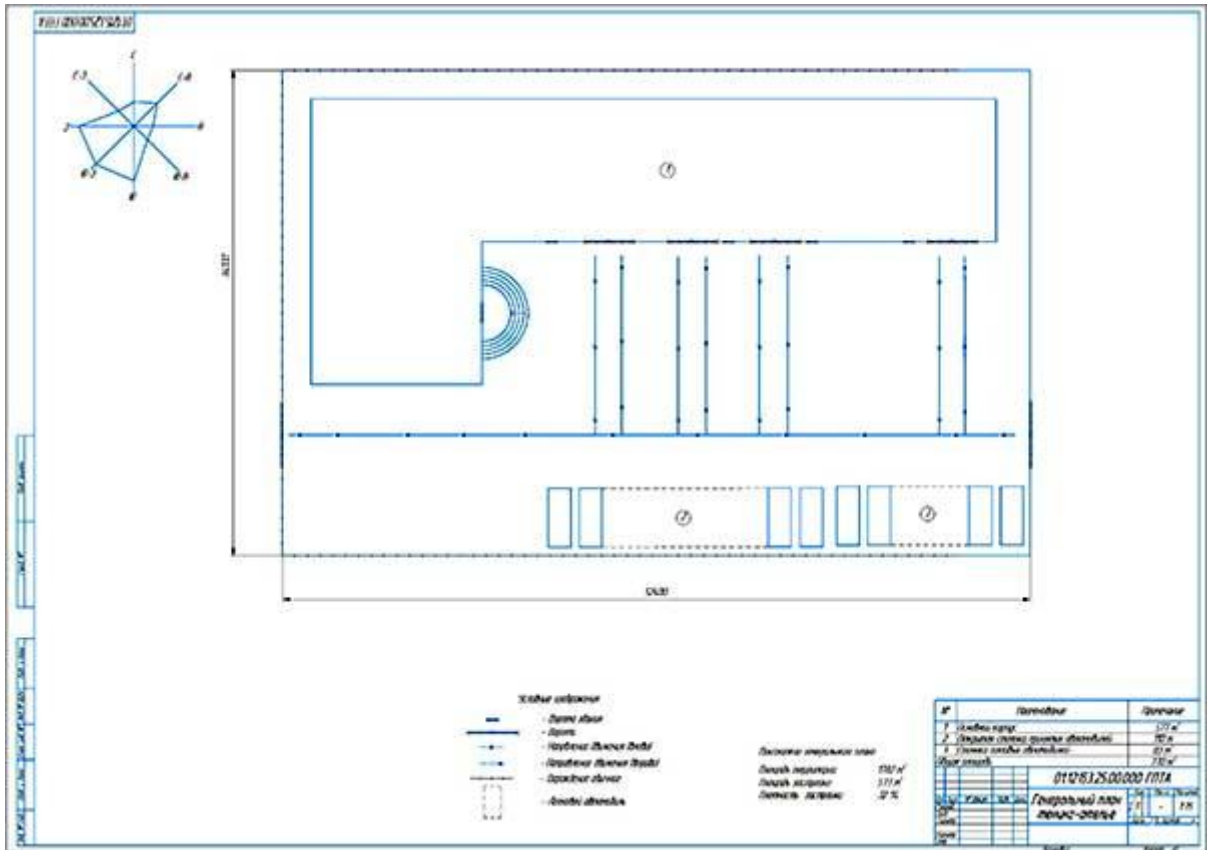


Рисунок 2.2 – План будівлі «CarRepairServices»

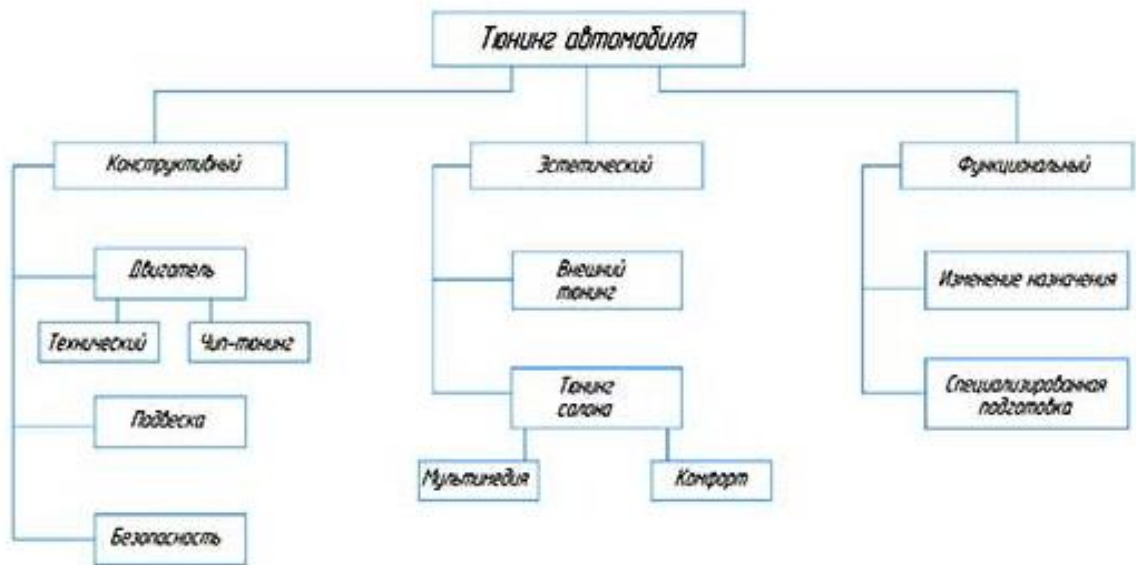


Рисунок 2.3 – Структура ателье «CarRepairServices»

2.2 Розробка програмного забезпечення

Технологія створення інформаційної системи[11] досить формалізована. Розробка інформаційної системи відноситься до керованих процесів, результат яких заздалегідь передбачуваний. Процес створення інформаційної системи складається з певної послідовності етапів.

Традиційно виділяють такі етапи створення інформаційної системи:

- перший етап – постановка задачі;
- другий етап – розробка та затвердження дизайну інформаційної системи;
- третій етап – здійснення інтеграції дизайну інформаційної системи з системою управління;
- четвертий етап – це процедура публікації інформаційної системи в мережі Інтернет;
- п'ятий і шостий етап – процес заповнення інформаційної системи необхідною інформацією і подальша його підтримка;

2.3 Контекстна діаграма

Кожна модель повинна мати контекстну діаграму верхнього рівня, на якій об'єкт опису представлений поодиноким блоком або оточеним дугами. Дуги на данній діаграмі пов'язують об'єкт з навколишнім середовищем. Контекстна діаграма встановлює область або межі моделювання. Так як контекстна діаграма представляє весь об'єкт, то її блок повинен мати саме загальне ім'я. Те ж відноситься і до імен інтерфейсних дуг. На контекстній діаграмі повинні бути вказані точка зору моделі та мета її створення.

Найбільш важливі особливості системи представлені в ієрархії, оскільки функція верхнього рівня декомпозирується на підфункції.

Поділ об'єкта на його структурні частини називається декомпозицією.

Підфункції, у свою чергу, декомпозируються далі, поки не будуть показані всі деталі, важливі з цієї точки зору. Кожна підфункція зображується окремим блоком, який є дочірньою деталізацією батьківського блоку на наступному (більш низькому) рівні. Області опису дочірніми діаграмами не повинні виходити за межі опису контекстної діаграми верхнього рівня.

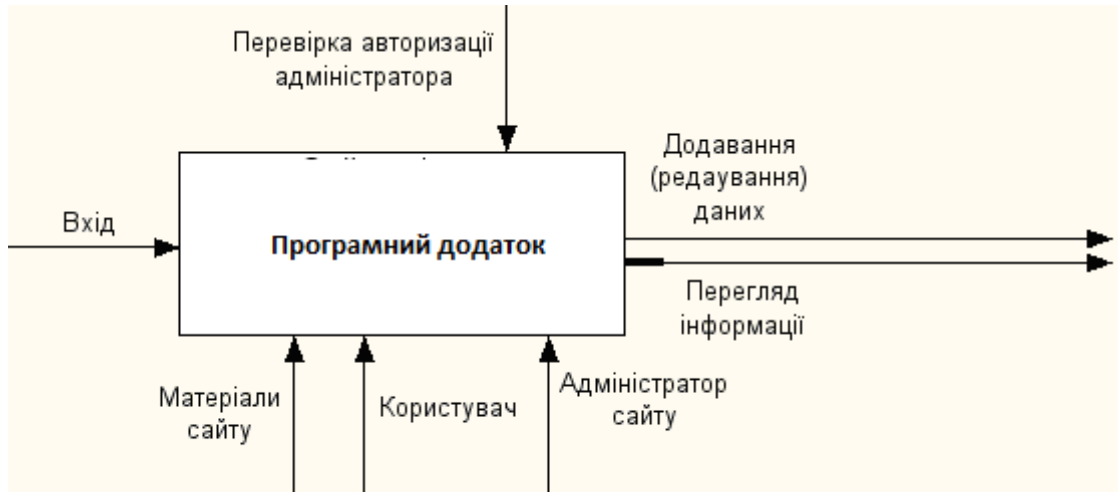


Рисунок 2.4– Контекстна діаграма

2.4 Діаграма декомпозиції

Основу методології IDEF0 складає графічна мова опису бізнес-процесів. Модель в нотації IDEF0 являє собою сукупність ієрархічно впорядкованих і взаємопов'язаних діаграм. Вершина цієї деревовидної структури, що представляє собою саме загальний опис системи та її взаємодії із зовнішнім середовищем, називається контекстною діаграмою. Після опису системи в цілому проводиться розбиття її на великі фрагменти. Цей процес називається функціональною декомпозицією, а діаграми, які описують кожен фрагмент і взаємодію фрагментів, називаються діаграмами декомпозиції.

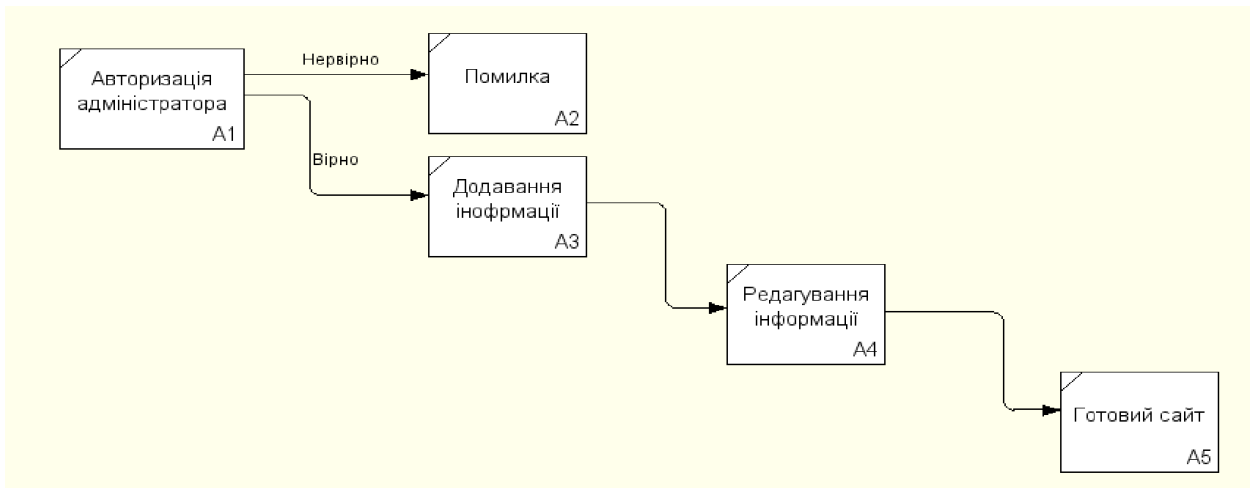


Рисунок 2.5– Діаграма декомпозиції

2.5 Діаграма прецедентів

Діаграма прецедентів – в UML, діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть даної діаграми полягає в наступному: проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (англ. use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

- асоціації (англ. association relationship);
- включення (англ. include relationship);
- розширення (англ. extend relationship);
- узагальнення (англ. generalization relationship);

При цьому загальні властивості варіантів використання можуть бути представлені трьома різними способами, а саме – за допомогою відношень включення, розширення і узагальнення.

Відношення асоціації – одне з фундаментальних понять у мові UML і в тій чи іншій мірі використовується при побудові всіх графічних моделей систем у формі канонічних діаграм.

Включення (англ. include) у мові UML – це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. При цьому відношенням залежності (англ. dependency) є таке відношення між двома елементами моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного).

Відношення розширення (англ. extend) визначає взаємозв'язок базового варіанта використання з іншим варіантом використання, функціональна поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов.

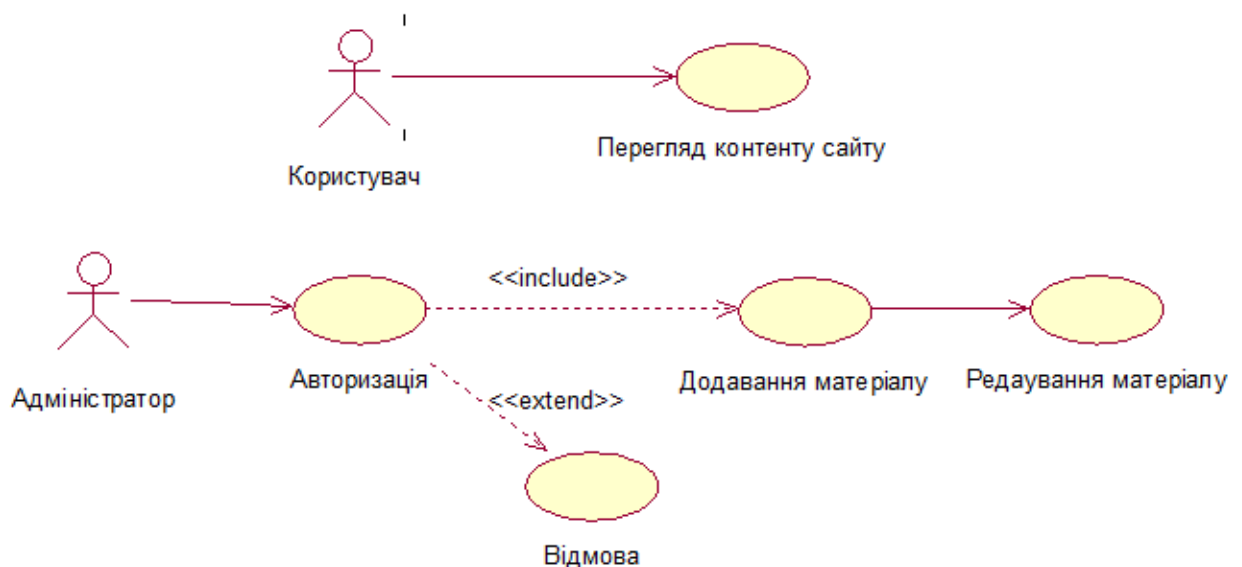


Рисунок 2.6– Діаграма прецедентів

2.6 Діаграма діяльності

Діаграма діяльності – в UML, візуальне представлення графу[14] діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Дія (англ. action) є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності. Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Діаграми активностей будуються з обмеженої кількості фігур, з'єднаних стрілочками. Найважливіші типи фігур:

- скруглені прямокутники позначають дії;
- ромби позначають рішення;
- риски позначають початок (розподіл) чи кінець (об'єднання) паралельних активностей;
- чорний кружок позначає старт (початковий стан) процесу;
- чорний кружок в колі позначає кінець (кінцевий стан);

Стрілки ведуть від старту до кінця і позначають порядок в якому відбуваються активності.

Діаграма активностей може вважатись формою блок-схеми.

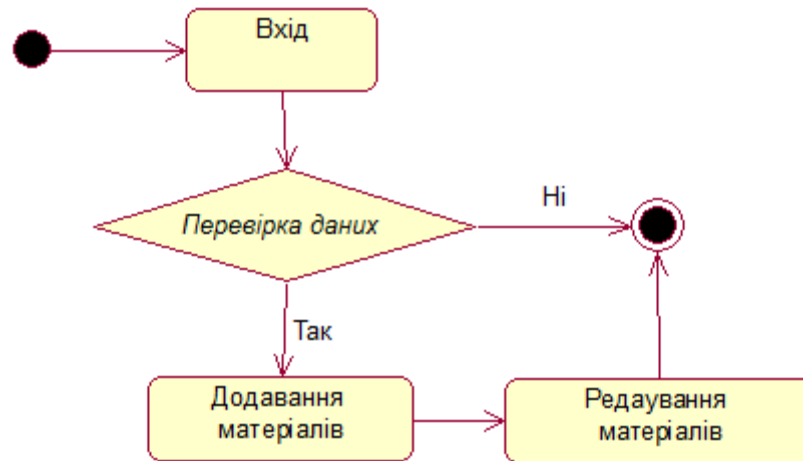


Рисунок 2.7– Діаграма діяльності

2.7 Розробка ER-моделі предметної області

Головними поняттями ER-моделі є сутність, зв'язок і атрибут. У діаграмах ER-моделі сутність подається у вигляді прямокутника, що містить ім'я сутності. При цьому ім'я сутності – це ім'я типу, а не деякого конкретного екземпляра цього типу. Для кращого розуміння ім'я сутності може супроводжуватися прикладами конкретних об'єктів цього типу.

Кожний екземпляр сутності повинен відрізнитися від будь-якого іншого екземпляра тієї ж сутності (ця вимога до повної міри аналогічна вимозі відсутності кортежів-дублікатів у реляційних таблицях).

Зв'язок – це асоціація, що графічно зображується та установлюється між двома сутностями. Ця асоціація завжди є бінарною і може існувати між двома різними сутностями або між сутністю і нею ж самою (рекурсивний зв'язок). У будь-якому зв'язку виділяються два кінці (відповідно до існуючої пари сутностей, що зв'язуються), на кожному з яких вказується ім'я кінця зв'язку, ступінь кінця зв'язку (скільки екземплярів даної сутності зв'язується), обов'язковість зв'язку (тобто чи будь-який екземпляр даної сутності повинен брати участь у даному зв'язку). Зв'язок подається у вигляді лінії, що зв'язує дві сутності або веде від сутності до неї ж самої. Про це в місці "стикування" зв'язку із сутністю вказує триточковий вхід у

прямокутник сутності, якщо для цієї сутності в зв'язку можуть використовуватися багато (many) екземплярів сутності, і одноточковий вхід, якщо в зв'язку може брати участь тільки один екземпляр сутності. Обов'язковий кінець зв'язку зображується суцільною лінією, а необов'язковий – переривчастою лінією.

В даній інформаційній системі зв'язки один до одного (рис. 2.8).

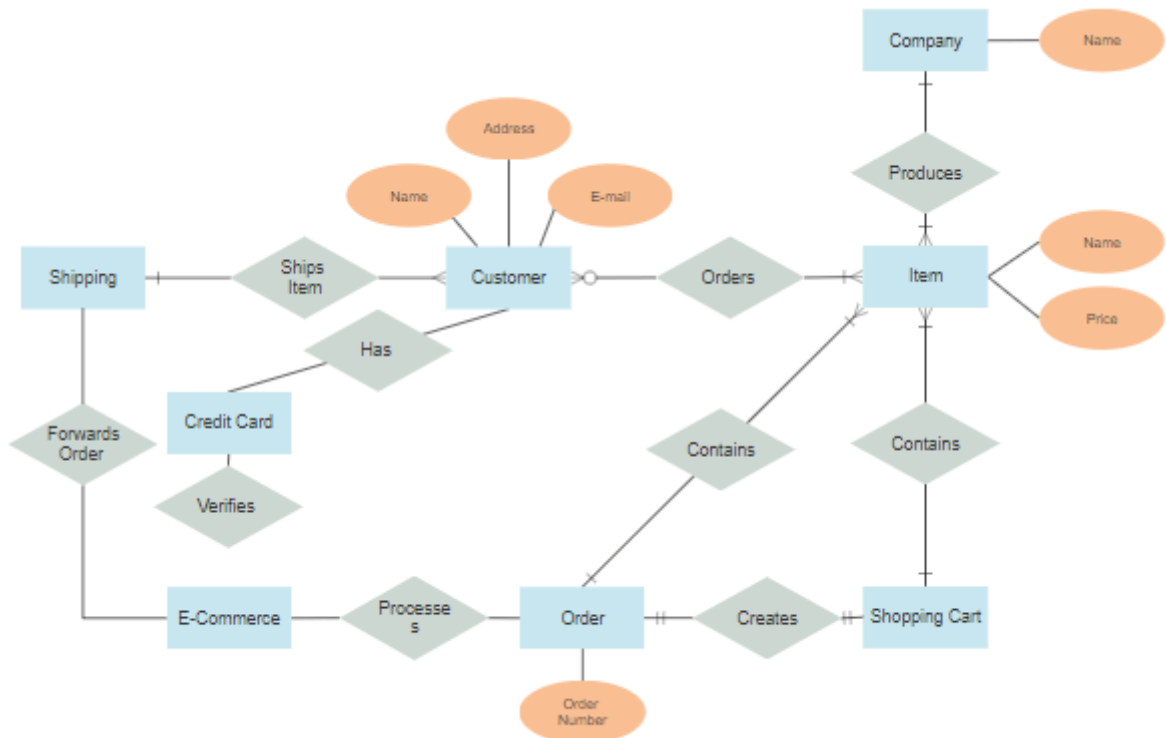


Рисунок 2.8– ER-модель або схема даних

2.8 Структура інформаційної системи

Інформаційна структура має важливе значення при розробці. Тому розробці інформаційної структури слід приділити особливу увагу.

Інформаційна структура – це спосіб організації інформаційних матеріалів системи разом зі всіма зв'язками, що дозволяє цим матеріалам взаємодіяти один з одним. І від того, наскільки добре організована така взаємодія, залежить рівень комфорту при використанні програмного забезпечення.

При розробці інформаційної структури інформаційної системи слід врахувати, що в залежності від кількості інформаційних матеріалів, які потрібно розмістити на системі, модель інформаційної системи може бути лінійною, ієрархічною і заплутаною («павутина»). Кожна з таких моделей організації інформаційної системи буде мати свої переваги.

Ієрархічна структура («дерево») – це популярна модель організації для бізнесу, таких як сайти – представництва, корпоративні сайти. Ієрархічна структура сайту дозволяє відвідувачам керувати глибиною відвідування сайту.

Інформаційна структура у вигляді «дерева» може містити розділи, що діляться на категорії, які в свою чергу поділяються на підкатегорії. В класичному «дереві» один інформаційний матеріал може належати тільки одній підкатегорії, категорії або розділу. Інформаційна структура часто включає в себе стандартні розділи і категорії без урахування специфіки.

Враховуючи всі критерії до створення інформаційної структури програми, було вирішено використовувати ієрархічну структуру (Рис. 2.9), оскільки вона є найдоцільніша в даному випадку.

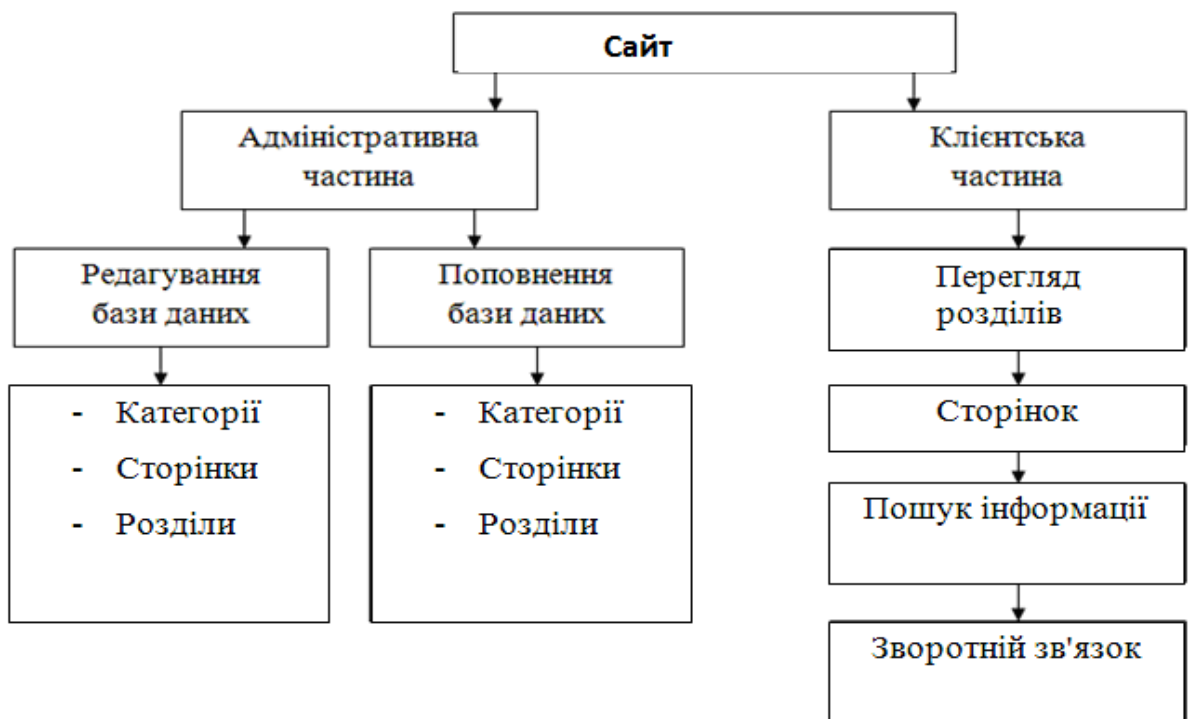


Рисунок 2.9 – Інформаційна структура

2.9 Вибір серверної мови програмування

При використанні програми написаної на серверній мові відбувається наступне – користувач дає запит на яку-небудь сторінку (переходить на неї за посиланням або вводить адресу в адресному рядку свого браузера) викликана сторінка спочатку обробляється на сервері[14], тобто виконуються всі програми, пов'язані зі сторінкою, і тільки потім повертається до відвідувача по мережі у вигляді файлу. Графічно можна показати це наступним чином(рис.2.10). Робота програм вже повністю залежна від сервера, на якому розташований сайт, і від того, яка версія тієї чи іншої мови підтримується. Важливою стороною роботи серверних мов є можливість організації безпосередньої взаємодії з системою управління базами даних (або СУБД) – сервером, на якому упорядковано зберігається інформація, яка може бути викликана в будь-який момент.

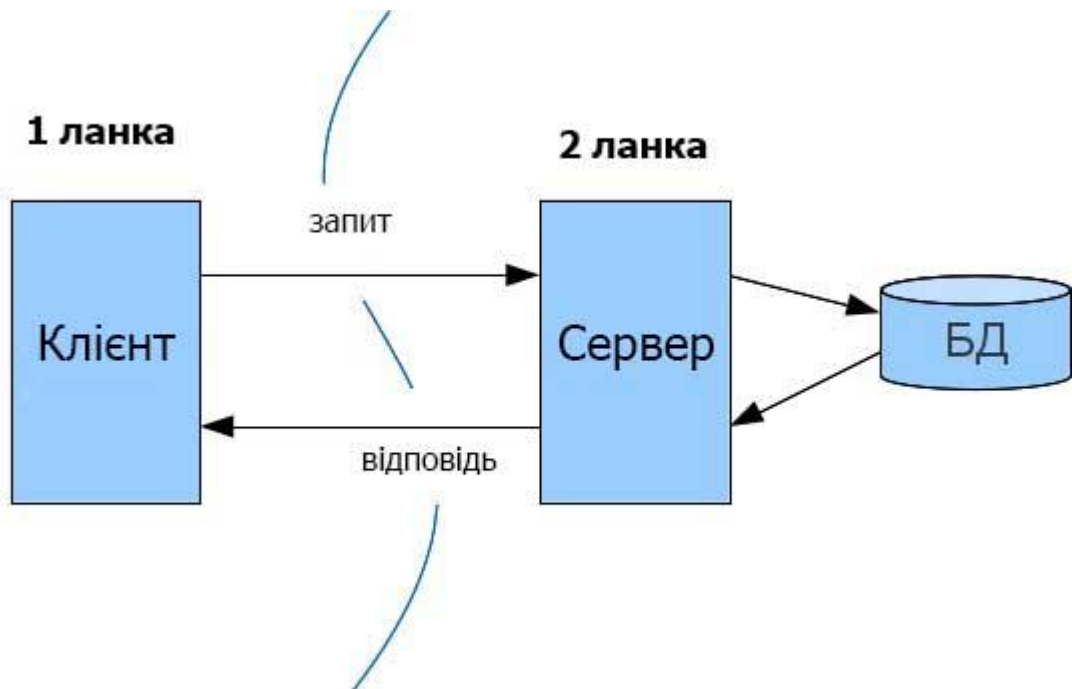


Рисунок 2.10 – Мережева організація

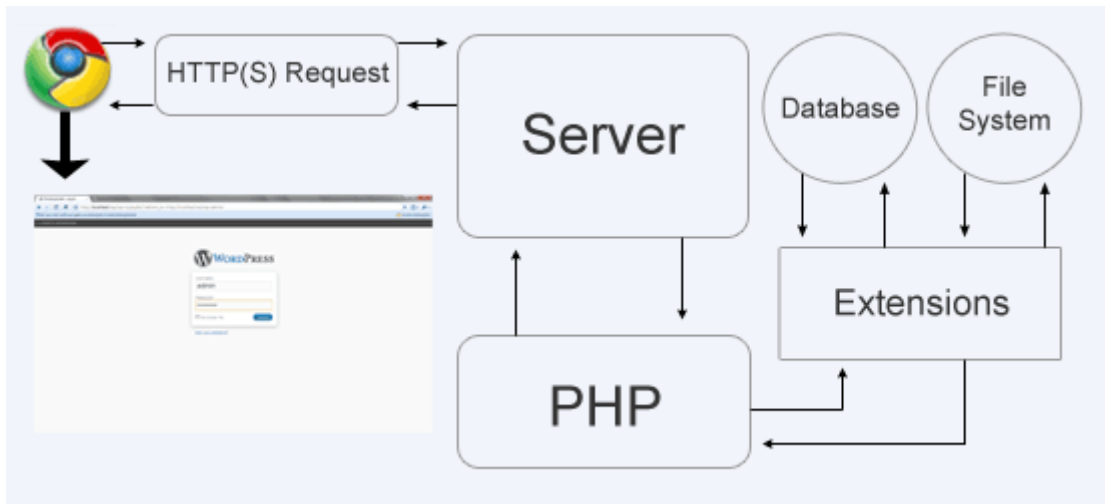


Рисунок 2.11–Схема роботи мови PHP

Рационально буде використовувати PHP оскільки вона зручна і підтримується більшістю хостинг – серверів. PHP (гіпертекстовий препроцесор) – скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервер. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок. PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта.

2.10 UML діаграми

В UML взаємозв'язок – зв'язок між елементами моделі. Зв'язок UML – це тип елементу моделі, який додає семантику до моделі, визначаючи структуру та поведінку між елементами моделі. Відносини UML [25] згруповані за такими категоріями:

- ребра діяльності – представляють потік між діями;
- асоціації вказують, що екземпляри одного елемента моделі пов'язані з екземплярами іншого елемента моделі;
- залежності вказують, що зміна одного елемента моделі може вплинути на інший елемент моделі;

- узагальнення вказують, що один елемент моделі є спеціалізацією іншого елемента моделі;
- реалізації вказують, що один елемент моделі забезпечує специфікацію, яку реалізує інший елемент моделі;

Переходячи до представлення змін в стані де ви можете встановлювати властивості та використовувати ключові слова для створення варіантів цих взаємозв'язків.

Асоціації

У моделях UML асоціація – це взаємозв'язок між двома класифікаторами, такими як класи або випадки використання, що описує причини взаємозв'язку та правила, що регулюють відносини.

Асоціація являє собою структурний зв'язок, який пов'язує два класифікатори. Як і атрибути, асоціації реєструють властивості класифікаторів. Наприклад, у відносинах між класами ви можете використовувати асоціації, щоб показати рішення щодо проектування, які ви прийняли щодо класів у вашій програмі, що містять дані, і показати, яким із цих класів потрібно обмінюватися даними. Ви можете використовувати функцію навігації асоціації, щоб показати, як об'єкт одного класу отримує доступ до об'єкта іншого класу або, у рефлексивній асоціації, до об'єкта того самого класу.

Назва асоціації описує характер взаємозв'язку між двома класифікаторами і повинна бути дієсловом або фразою.

У редакторі діаграм асоціація виглядає як суцільна лінія між двома класифікаторами

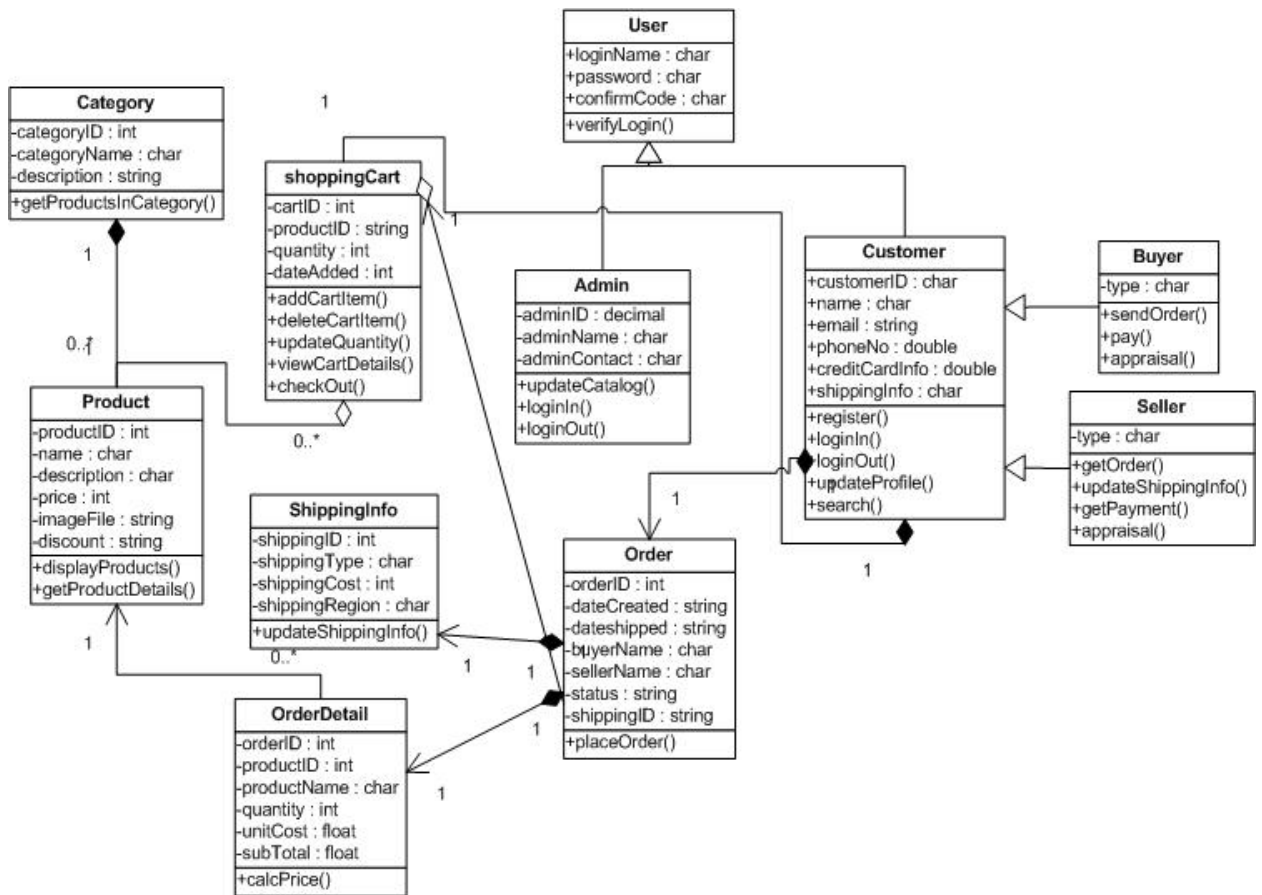


Рисунок 2.12–UMLдіаграми

2.11 Вибір локального сервера.

Після детального аналізу локальних серверів був вибраний Open Server.

Open Server – це портативна серверна платформа і програмне середовище, створена спеціально для веб-розробників з урахуванням їх рекомендацій і побажань. Програмний комплекс має багатий набір серверного програмного забезпечення, зручний, багатофункціональний продуманий інтерфейс, володіє потужними можливостями з адміністрування та налаштування компонентів. Платформа широко використовується з метою розробки, налагодження і тестування веб-проектів, а так само для надання веб-сервісів в локальних мережах.

Насамперед хочу зазначити, що Open Server – це цілком і повністю портативний сервер. Ніяких системних сервісів, куп сміття в реєстрі і system32. Ви можете всюди тягати його з собою на флешці (бажано на швидкісній), запускати на робочій / домашній машині без побоювання що у вас щось не запрацює.

У разі відсутності на комп'ютері потрібних системних компонентів Open Server встановить їх сам, досить вибрати в меню [Інструменти – Перший запуск] якщо сервер запускається на комп'ютері вперше.

Немає необхідності описувати основні можливості програми, оскільки в більшості своїй вони стандартні для такого роду софту. Самою собою що за допомогою Open Server можна запустити / зупинити сервер або відкрити потрібний домен. Набагато цікавіше буде дізнатися про специфічні «фічах», які роблять Open Server особливим, дійсно особливим:

- докладний перегляд логів всіх компонентів в реальному часі;
- вибір HTTP, СУБД і PHP модулів в будь-якому поєднанні;
- підтримка SSL і кириличних доменів з коробки;
- підтримка алиасов або по іншому доменних покажчиків, а так само зручна форма їх налаштування (привіт любителям мультісайтінга в Drupal!);
- створення локального піддомена без втрати видимості основного домену в мережі інтернет;
- доступ до доменів (в один клік) і швидкий доступ до шаблонів конфігурації модулів;
- багатомовний інтерфейс (Російська, Українська, Білоруська, Англійська);

Програма постійно удосконалюється, всі адекватні прохання з боку користувачів Open Server детально вивчаються і більшість з них реалізується!

2.12 Вибір візуального середовища програмування.

Візуальне середовище програмування – це інтегроване середовище розробки програмних засобів, яке містить редактор вихідного коду, компілятор чи/або інтерпретатор, засоби автоматизації збіркита засоби для спрощення розробки графічного інтерфейсу користувача. Середовища для візуального програмування також надають змогу конструювати програми шляхом оперування графічними об'єктами. Багато сучасних візуальних середовищ програмування використовуються для реалізації принципів об'єктно-орієнтованого підходу у розробці програмного забезпечення.

Візуальним середовищем програмування був вибраний PHPStorm (рис. 2.13). PHPStorm – комерційне крос-платформове інтегроване середовище розробки для PHP, яке розробляється російською компанією JetBrains на основі платформи IntelliJ IDEA.

PhpStorm розроблений на основі платформи IntelliJ IDEA, написаної на Java. Користувачі можуть розширити функціональність середовища розробки за рахунок установки плагінів, розроблених для платформи IntelliJ, або написавши власні плагіни.

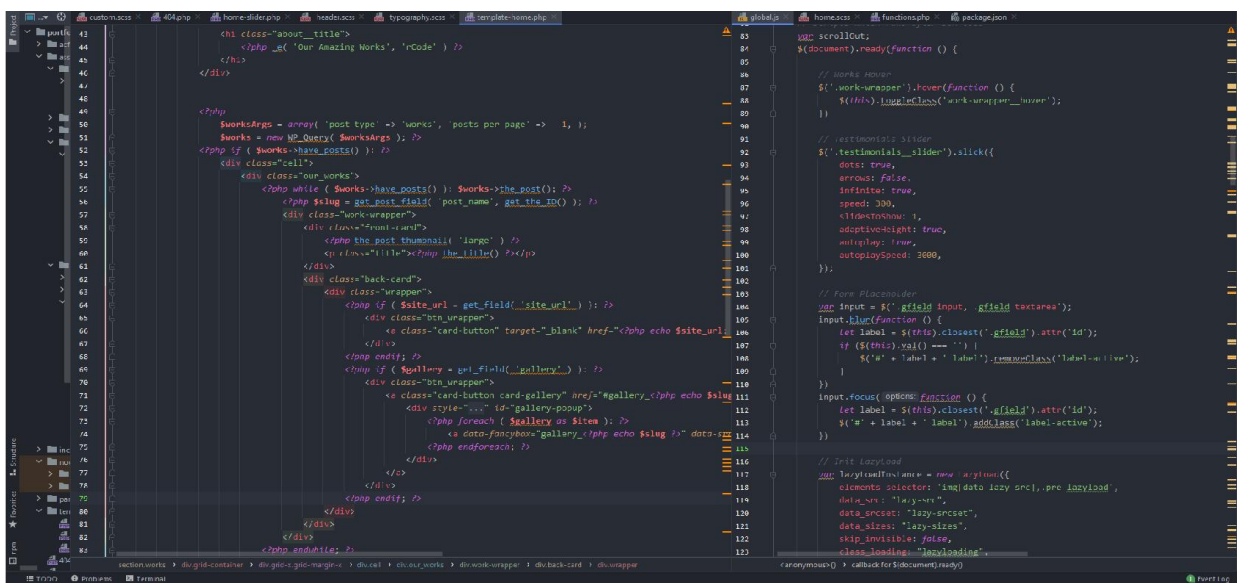


Рисунок 2.13 – Візуальне середовище програмування PHPStorm.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ

3.1 Розробка структури

Проаналізувавши переглянуті дані та схожі сайти була створена наступна структура сторінок сайту:

- головна сторінка;
- категорії товарів;
- живий пошук по сайту;
- жива фільтрація товарів;
- панель управління адміністратора;
- детальний огляд запчастини;

В свою чергу панель управління адміністратора має наступну структуру:

- додавання нових товарів;
- оповіщення про нові акції;
- додавання новин;
- додавання нових адміністраторів;
- перегляд замовлень;
- відправка повідомлень про надання або відмову в наданні

послуги;

Горизонтальне навігаційне меню сайту містить 5 основних пунктів, це: Головна, Допомога, Контакти, Наші роботи, Магазин.

3.2 Вигляд зі сторони користувача

Головна сторінка котра складається з 7 секцій. Такі як привітальний банер, послуги, як це працює, відгуки та список автомобілів з котрими ми працюємо.

На рис. 3.1 представлено головну сторінку з послугами

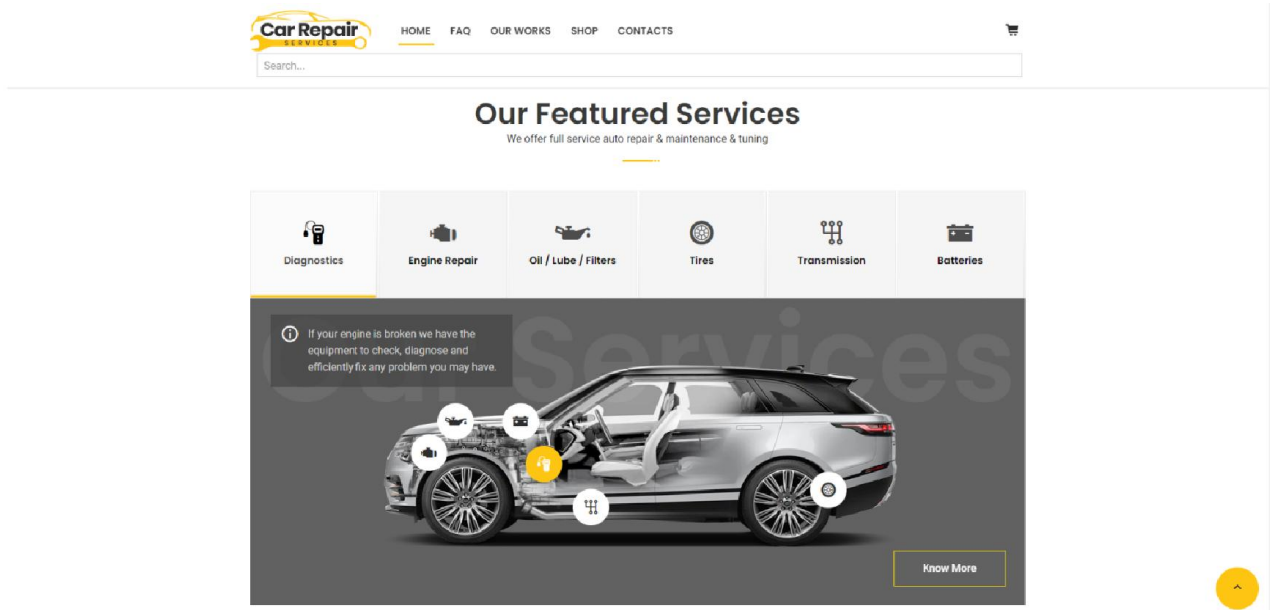


Рисунок 3.1 – Головна сторінка.

На наступному рисунку (рис. 3.2) сторінка наших робіт с елементами фільтрації, на котрій відображено автомобіль то тюнінгу та після

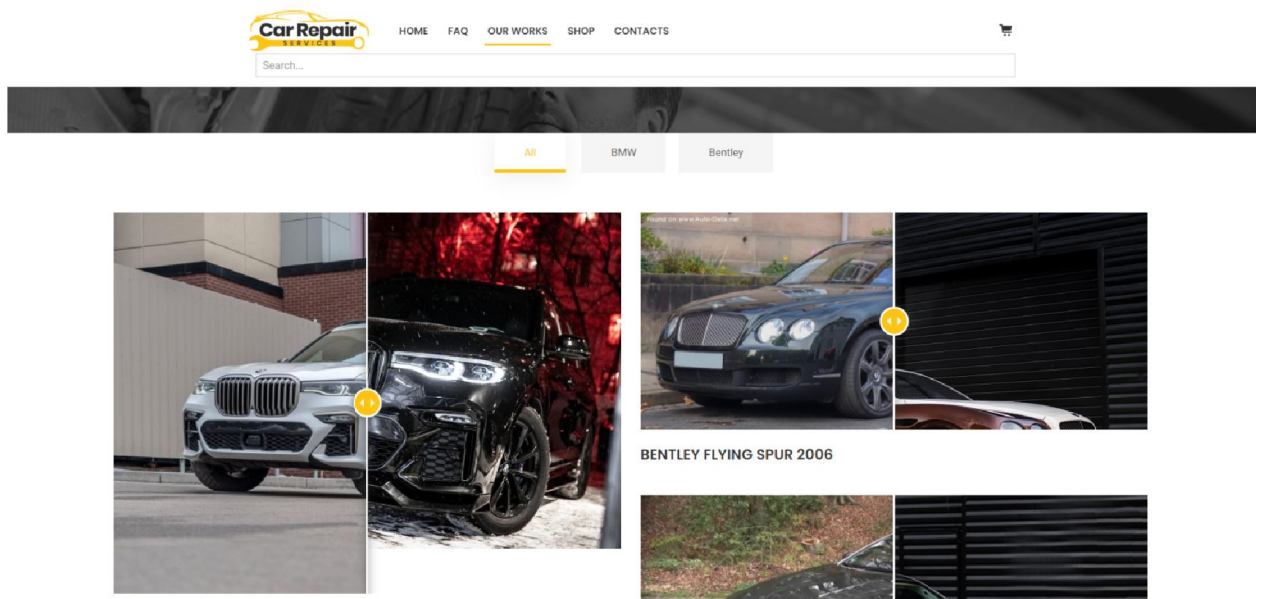


Рисунок 3.2 – Сторінка наших робіт.

Сторінка товарів відображена на рис.3.3, де ви можете відфільтрувати товари по критеріям

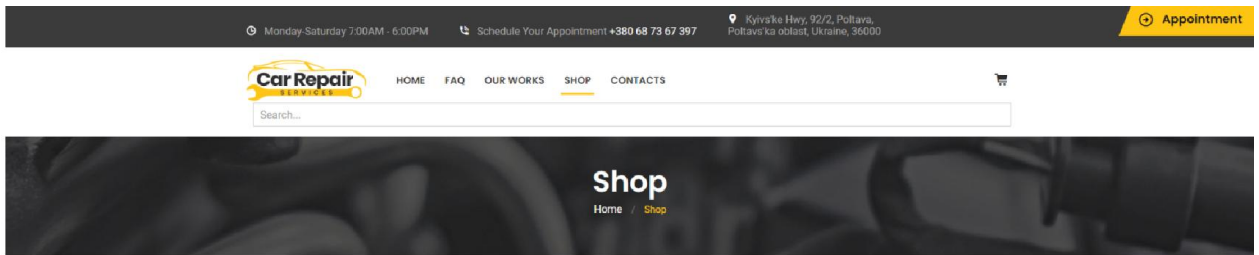


Рисунок 3.3 – Сторінка магазину

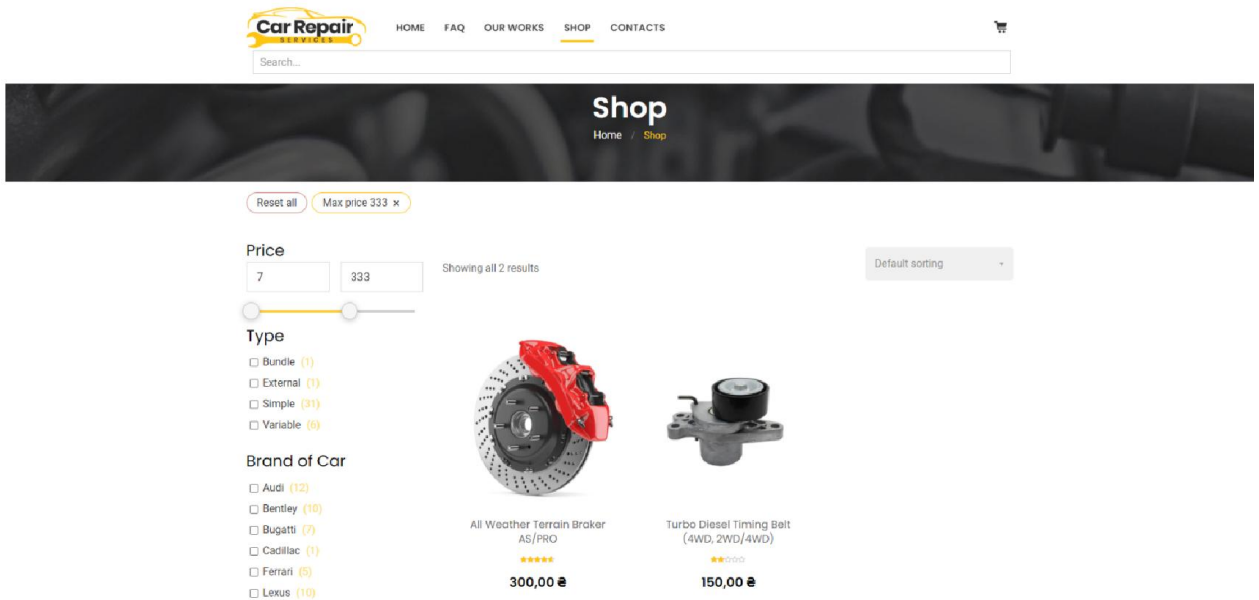


Рисунок 3.4 – Сторінка магазину з використаним фільтром

На рис.3.5 відображено спливаюче вікно з формою запису на прийом доспеціалістів.

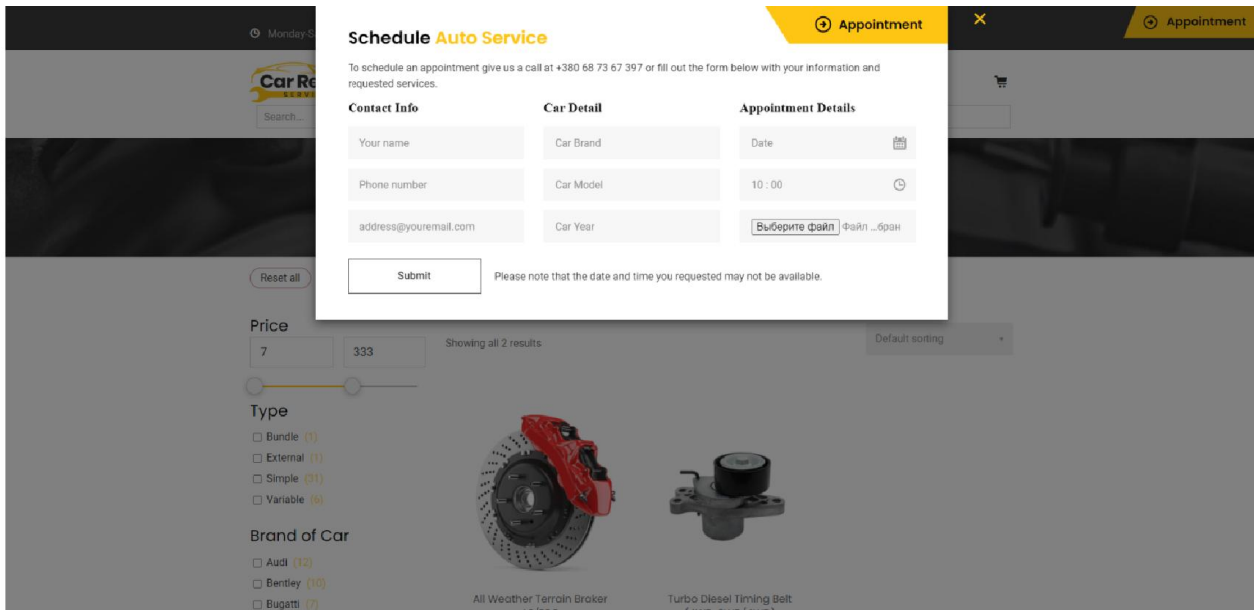


Рисунок 3.5 – Спливаюче вікно.

Після відправки форми на пошту адміністратора приходять повідомлення про запис, який відображено на малюнку 3.6

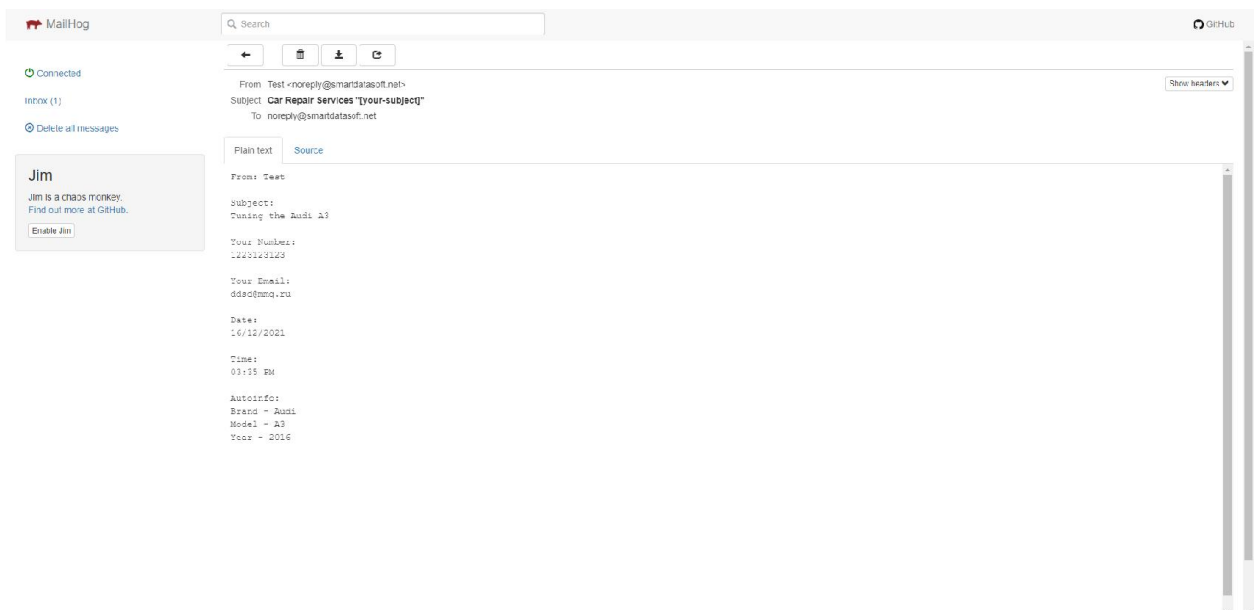


Рисунок 3.6 – Шаблон повідомлення.

Після додавання товару до корзини, ви можете продовжити покупку, або почати оформлювати замовлення через “форму замовлення” яка представлена на рис.3.7

Рисунок 3.7 – Форма замовлення

Інтелектуальний пошук робить пошук лише по товаром(рис. 3.8), так же можна налаштувати для пошуку по всій системи

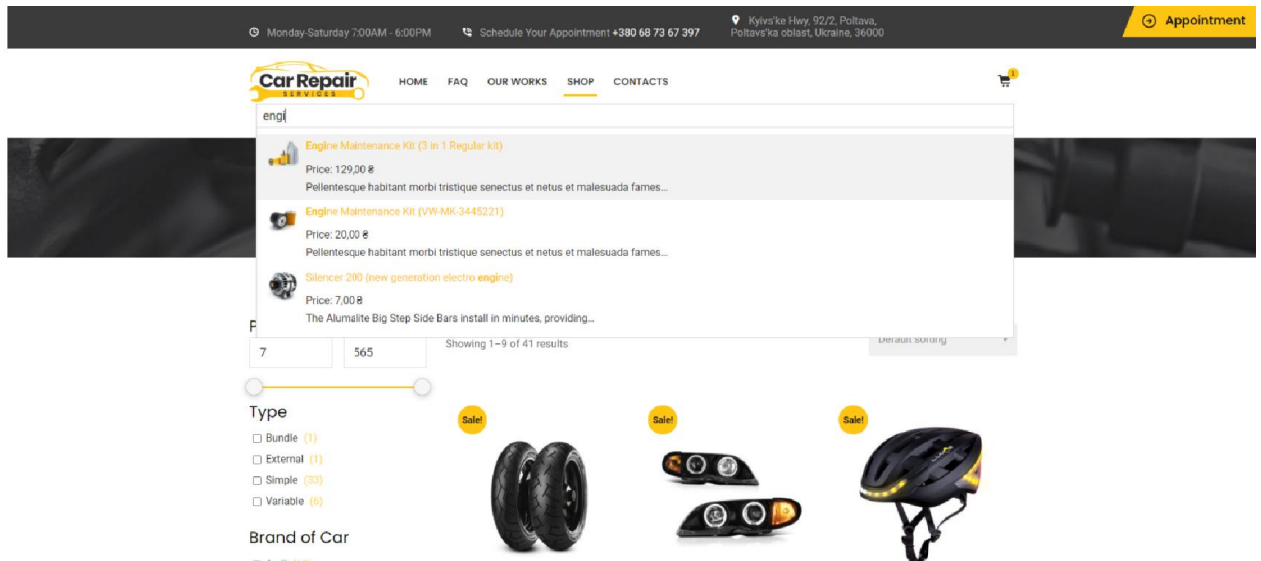


Рисунок 3.8 – Пошук інформації.

У користувачів з роллю Адміністратор є доступ до адмін панелі (рис. 3.9)

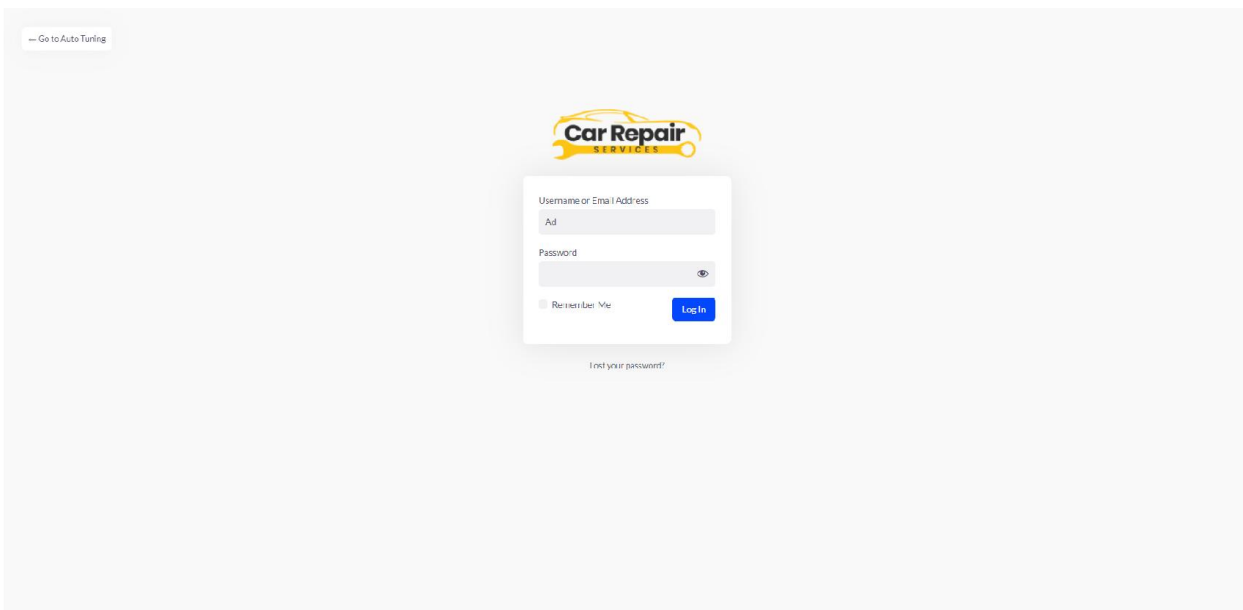


Рисунок 3.9 – Вхід до панелі адміністрування.

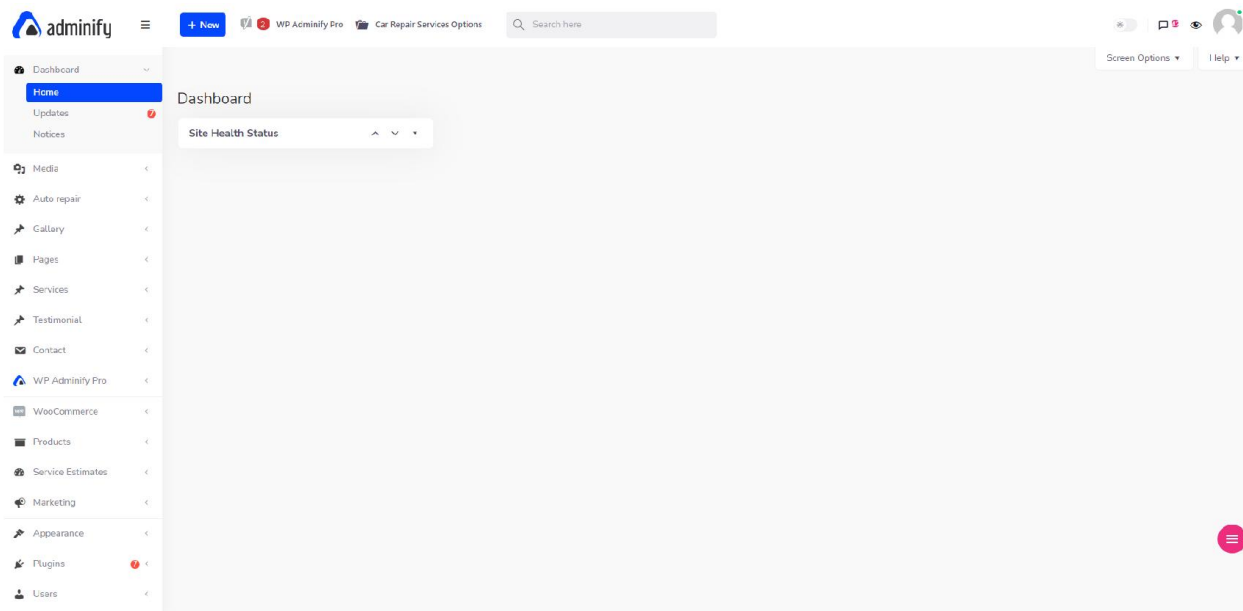


Рисунок 3.10 – Панель адміністрування.

Для додавання нового продукту потрібно зайти в адмін панель та перейти в пункт Products, де будуть відображатись всі продукти та можна додати новий (рис. 3.11)

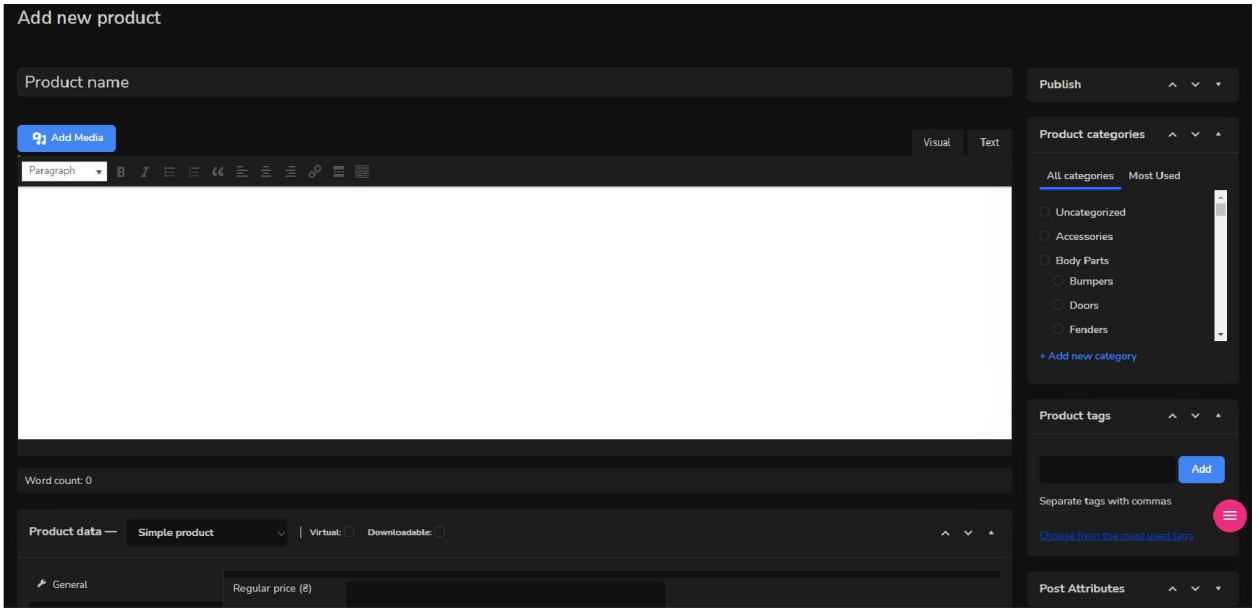


Рисунок 3.11 – Додавання нового товару.

Після оформлення замовлення, заказ потрапляє до розділу замовлення після чого співробітник може переглянути його та перетелефонувати якщо це необхідно (рис.3.12)

Order	Date	Status	Total
#1680 jakjfskdfjdf fdfsfdfdfsf	Oct 11, 2019	Cancelled	15,00 \$
#1675 Manuel Tenora	Sep 11, 2019	Processing	20,00 \$
#1666 11 22	May 25, 2019	Cancelled	12,00 \$
#1662 A A	May 7, 2019	Cancelled	53,00 \$
#1661 VEERENDRA SINGH	May 3, 2019	On hold	12,00 \$
#1655 ibk barry	Mar 22, 2019	Processing	30,00 \$
#1654 ibk barry	Mar 22, 2019	Processing	65,00 \$
#1653 вцяу вйцвйцвйцв	Mar 6, 2019	Processing	60,00 \$
#1636	Dec 8, 2018	Processing	0,00 \$
#3687 gustavo consentinio consentino	Jul 28, 2018	Cancelled	797,00 \$

Рисунок 3.12 – Перегляд замовлень.

РОЗДІЛ 4

ТЕСТУВАННЯ

4.1 Тестування ІС

Якість програмного продукту характеризується набором декількох властивостей, що визначають, наскільки продукт «якісний» з точки зору зацікавлених сторін, таких як замовник продукту, спонсор, кінцевий користувач, розробники і тестувальники продукту, інженери підтримки, співробітники відділів маркетингу, навчання і продажів. Кожен з учасників може мати різне уявлення про продукт і те, наскільки він хороший чи поганий, тобто про те, наскільки висока якість продукту. Таким чином, постановка задачі забезпечення якості продукту виливається у завдання визначення зацікавлених осіб, їх критеріїв якості і потім знаходження оптимального рішення, що задовольняє цим критеріям. Тестування є одним із головних способів забезпечення якості розробки програмного забезпечення і входить в набір ефективних засобів сучасної системи забезпечення якості програмного продукту.

З технічної точки зору тестування ІС полягає у відповідності макету з готовим результатом та коректності роботи.

4.2 Тест-план

Тест план-це документ описує весь обсяг робіт з тестування, починаючи з опису об'єкта, стратегії, розкладу, критеріїв початку і закінчення тестування, до необхідного в процесі роботи обладнання, спеціальних знань, а також оцінки ризиків з варіантами їх вирішення.

Тест план повинен як мінімум відповідати на такі питання:

- що треба тестувати (об'єкт тестування: система, додаток, обладнання);

- що будете тестувати (список функцій і компонент тестується системи);
- як будете тестувати (стратегія тестування – види тестування і їх застосування по відношенню до тестованого об’єкту);
- коли будете тестувати (послідовність проведення робіт: підготовка, тестування, аналіз результатів, в розрізі запланованих фаз розробки проекту);
- критерії початку і закінчення тестування;

Тестування проводилося в 3 найпопулярніших браузерях. Результати проведення фінального тестування верстки зображена в таблиці 4.1

Таблиця 4.1 – Тестування верстки

	Mozilla Firefox	Google Chrome	Microsoft Edge
Коректне відображення картинок	Успішно	Успішно	Успішно
Клікабельність посилань	Успішно	Успішно	Успішно
Відображення колірної гама всіх елементів	Успішно	Успішно	Успішно
Коректність скролла	Успішно	Успішно	Успішно
Коректне розміщення банерів	Успішно	Успішно	Успішно
Коректне масштабування сторінок	Успішно	Успішно	Успішно
Відображення в різних розширеннях	Успішно	Успішно	Успішно
Коректна зміна розміру текстових полів	Успішно	Успішно	Успішно

Після завершення тестування верстки було проведено тестування функціоналу. Результати тестування наведені в таблиці 4.2.

Таблиця 4.2 – Тестування функціоналу

Клікабельність кнопок	Успішно	Успішно	Успішно
Коректне відображення картинок	Успішно	Успішно	Успішно
Меню	Успішно	Успішно	Успішно
Локалізація	Успішно	Успішно	Успішно
Клікабельність посилань	Успішно	Успішно	Успішно
Коректне заповнення бланка зворотнього зв'язку (перевірка Email)	Успішно	Успішно	Успішно

Також було розроблено та виконано тест-кейс, для перевірки роботи системи. Виконані тест кейси наведені в таблиці 4.3.

Таблиця 4.3 – Тест-кейси

Тест-кейс №1. Тестування при невірному введенні логіну	
Дії	Очікуваний результат
1. Відкриваємо інформаційну систему від імені адміністратора	Редірект на сторінку з формою авторизації
2. Вводимо незареєстрований логін	Логін введений
3. Натискаємо кнопку "Вхід"	З'явилося повідомлення про те, що логін невірний
Тест-кейс №2. Тестування при невірному введенні пароля	
Дії	Очікуваний результат
1. Відкриваємо інформаційну систему від імені адміністратора	Редірект на сторінку з формою авторизації
2. Вводимо зареєстрований логін	Логін введений
3. Вводимо невірний пароль	Пароль введений
4. Натискаємо кнопку "Вхід"	З'явилося повідомлення про те, що пароль невірний

Продовження таблиці 4.3

Тест-кейс №3. Тестування при вірному введенні логіна та пароля	
Дії	Очікуваний результат
1. Відкриваємо інформаційну систему від імені адміністратора	Редірект на сторінку з формою авторизації
2. Вводимо зареєстрований логін	Логін введений
3. Вводимо вірний пароль	Пароль введений
4. Натискаємо кнопку "Вхід"	Вхід в інформаційну систему від імені адміністратора

Провівши тестування, були знайдені та усунені помилки. Інформаційна система працює коректно. Функціонал повністю та безпомилково виконує поставлені задачі.

ВИСНОВКИ

В даній дипломній роботі була спроектована та програмно реалізована інформаційна система з надання послуг тюнінгу автомобілів в Україні. В рамках дипломної роботи було досліджено та проаналізовано кілька існуючих інформаційних систем головних міст України. Було визначено деякі недоліки цих інформаційних систем та зроблено правильні висновки, аби запобігти повторюванню помилок.

Враховуючи основні вимоги до створення, розроблена мною інформаційна система забезпечить :

- можливість вибору товару;
- швидке та якісне обслуговування;
- простота доступу та замовлення;
- зручне керування адміністратора;

Головними перевагами даної інформаційної системи можна назвати гарний дизайн, який буде приваблювати та зацікавлювати користувачів. Зручна навігація по інформаційній системі завдяки асоціативному пошуку, дозволить швидко визначитися та замовити потрібну деталь. Класифікація товарів на 3 класи спростить пошуки.

Враховуючи вище сказане можна зробити висновок, що всі поставлені завдання в дипломній роботі виконані у повному обсязі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бібліотека навчальних матеріалів онлайн. – 2010. [Електронний ресурс] –
Режим доступу до ресурсу: https://pidru4niki.com/1222090547713/informatika/informatsiyeni_sistemi
2. Леонтьев Б.В. Web-Дизайн: Тонкости, хитрости и секреты / Леонтьев Б.В. – М. Майор, 2001.–170 с.
3. Популярные Web-сервисы: практика использования. Айверсон Уилл. КУДИЦ – ОБРАЗ, 2005.
4. Разработка Web – приложений с помощью PHP и MySQL, 3-е издание. Люк Веллинг, Лора Томсон. Вильямс, 2005.
5. Установа навчання VSAVM. – 2010. [Електронний ресурс] –
Режим доступу до ресурсу: <http://www.vsavm.by/knigi/kniga3/220.html>
6. Фролов А.В., Фролов Г.В. Глобальные сети компьютеров. Практическое введение в Internet, E-mail, FTP, WWW, и HTML, программирование для Windows Sockets. – Диалог – МИФИ, 1996. Ковалев А., Курдюмов И и др. Управление проектом по созданию интернет-сайта – М.: Альпина Паблишер, 2001. – 337 с.
7. Internet, Сервіси, HTML і web дизайн. Глинський. Деол, 2000.
8. WEB-дизайн Руководство пользователя. Под ред. Леонтьева. Познавательная книга, 2000.
9. Крамер Р. «HTML: наочний курс Web-дизайну»
10. Ботт Эд. Разработка веб-сайтов. – М., 2004.
11. Куперштейн В.И. Современные информационные технологии в делопроизводстве и управлении / В.И. Куперштейн. СПб.: БХВ – Санкт-Петербург, 2000. – 256 с.
12. Кушнарев Л.И., Хицков Е.А., Гальчич М.А. Методические рекомендации по дипломному проектированию. – М.: ФГОУ ВПО МГАУ., 2005. – 114 с.

13. Трофимов В.В. Информационные технологии: учебник. /Трофимов В.В – Москва: Издательство «ЮРАЙТ», 2011. – 624 с.
14. Martin Brampton. PHP5 CMS Framework Development./ Martin Brampton – PASCIT publishing, June 2008. – 328 p. – ISBN 978-1-847193-57-5
15. Горнаков С. Г., Осваиваем популярные системы управления сайтом (CMS)/ Горнаков С. Г – М.: ДМК Пресс, 2009 .– 336 с.
16. Микола Дроздов – Створи Свій Власний WordPress Сайт./ Микола Дроздов – 2011. – 31 с.
17. Люк Веллінг Розробка Web-додатків за допомогою PHP і MySQL/ Люк Веллінг, Лора Томсон Москва: Вільямс, 2005. – 876 с.
18. Кирсанов Веб-дизайн: Книга Дмитрия Кирсанова / Кирсанов, Дмитрий. – М.: СПб: Символ-Плюс, 2001. – 376 с
19. Вовк, Е.Т. Верстка для начинающих. PageMaker, Photoshop и CorelDraw / Е.Т. Вовк. – М.: КУДИЦ-Образ, 2006. – 400 с
20. Нильсен, Якоб Веб-дизайн. Книга Якоба Нильсена / Якоб Нильсен. – М.: Символ-плюс, 2006. – 512 с.
21. Леонтьева, Н.Н. Автоматическое понимание текстов: системы, модели, ресурсы / Н.Н. Леонтьева. – М.: Академия, 2006. – 304 с.
22. Дубаков Веб-мастеринг средствами CSS / Дубаков, Михаил. – М.: БХВ-Петербург, 2002. – 532 с.
23. Кедлек, Тим Адаптивный дизайн. Делаем сайты для любых устройств / Тим Кедлек. – М.: Питер, 2013. – 333 с
24. Гультияев, А. К. Проектирование и дизайн пользовательского интерфейса / А.К. Гультияев, В.А. Машин. – М.: Корона-Принт, 2010. – 350 с.
25. Ларман, Крэг Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку / Крэг Ларман. – М.: Вильямс, 2013. – 736 с.

ДОДАТОК А

КОД ПРОГРАМИ

```

<?php
$car_option = car_repair_services_options();
if(isset($car_option['is_modal_enable']) &&$car_option['is_modal_enable'] == 1){ ?>
<div class="modal fade" id="appointmentForm">
<div class="modal-dialog container">
<div class="modal-content">
<div class="modal-header"><a href="#" class="appointment" data-toggle="modal" data-
target="#appointmentForm"><i class="icon-shape icon"></i>
<?php
                if( isset( $car_option['car_repair_services-header-top-right-line'] )
&&$car_option['car_repair_services-header-top-right-line'] != '' ){
echo '<span>'. esc_html( $car_option['car_repair_services-header-top-right-line'] )
.'</span>';
                }
?></a>
<button type="button" class="close" data-dismiss="modal" aria-label="Close"><i class="icon-
close"></i></button>
</div>
<div class="modal-body">
<div class="container-fluid">
<h2 class="modal-title-main"><?php echo wp_kses_post( $car_option['modal_title'] ); ?></h2>
<?php echo wp_kses_post( $car_option['modal_content'] ); ?>
<div class="divider divider-sm"></div>
<?php echo sprintf(__('%s', 'car-repair-services'),
do_shortcode($car_option['modal_mc_form'])); ?>
</div>
</div>
</div>
</div>
</div>

<?php }

<?php
$style = $post->style;
$post_id = get_the_ID();
$client_name = get_post_meta( get_the_ID(), 'framework-client-name', true );
$client_designation = get_post_meta( get_the_ID(), 'framework-client-designation', true );
$attachement = wp_get_attachment_image_src( get_post_thumbnail_id( $post_id ), 'car-repair-
services-testimonial' );

```

```

$attachement1 = wp_get_attachment_image_src( get_post_thumbnail_id( $post_id ), 'car-repair-
services-testimonial-grid' );
?>

<?php

$car_repair_services = car_repair_services_options();
$theme = isset( $car_repair_services['theme_setting'] )
&&$car_repair_services['theme_setting'] == '1';

if ( $theme != '1' ) {
?>

<div class="col-item">
<div class="testimonial-item">
<div class="testimonial-item__content">
<?php the_content(); ?>
</div>
<div class="testimonial-item__footer">
<div class="testimonial__img">
">
</div>
<div class="testimonial__description">
<?php echo esc_html( $client_name ); ?>
<span><?php echo esc_html( $client_designation ); ?></span>
</div>
</div>
</div>
</div>

<?php
} else {
?>
<?php if ( $style ) { ?>
<div class="testimonial-card">
<input type="hidden" value="<?php echo get_the_ID(); ?>" />
<div class="testimonial-card-title">
<h5>
<?php
                if ( $client_name != '' ) {
echo esc_html( $client_name );
                }
?>

```

```

</h5></div>
<div class="testimonial-card-text"><?php echo the_content(); ?></div>
<div class="testimonial-card-rating">
<div class="rating rating-<?php echo esc_html( $client_rating ); ?>">
<i class="icon-star"></i><i class="icon-star"></i><i class="icon-star"></i><i class="icon-
star"></i><i class="icon-star"></i>
</div>
</div>
</div>
<?php } else { ?>
<div class="testimonials-item <?php echo esc_attr( $extra_class ); ?><?php
    if ( $i == 1 ) :
?>
testimonials-item--dark<?php endif; ?>">
<div class="inside">
<div class="meta">
<span class="rating"><i class="icon-star"></i><i class="icon-star"></i><i class="icon-
star"></i><i class="icon-star off"></i><i class="icon-star"></i></span><span class="username">
<?php
    if ( $client_name != '' ) {
echo esc_html( $client_name );
    }
?>
</span></div>
<div class="text"><?php echo the_content(); ?></div>
</div>
<div class="bg-image"
<?php
    if ( $attachement != array() ) {
?>
        style="background-image: url(<?php echo esc_url( $attachement[0]
); ?>);"
<?php
    }
?>
>
</div>
</div>
<?php
}
}

```



```

<?php
/**
 * Result Count
 */

if ( ! defined( 'ABSPATH' ) ) {
exit;
}

$car_repair_services = car_repair_services_options();
$theme               = isset( $car_repair_services['theme_setting'] )
&&$car_repair_services['theme_setting'] == '1';

if ( $theme != '1' ) {
?>
<div class="filters-row-left">
<span class="show-result">
<?php

    if ( 1 === $total ) {
        _e( 'Showing the single result', 'car-repair-services' );
    } elseif ( $total <= $per_page || -1 === $per_page ) {
printf( _n( 'Showing all %d result', 'Showing all %d results', $total, 'car-repair-services'
), $total );
    } else {
$first = ( $per_page * $current ) - $per_page + 1;
$last  = min( $total, $per_page * $current );
printf( _nx( 'Showing the single result', 'Showing %1$d&ndash;%2$d of %3$d results', $total,
'with first and last result', 'car-repair-services' ), $first, $last, $total );
    }
?>
</span>
</div>
<?php
} else {
?>

<span class="show-result">
<?php

    if ( 1 === $total ) {
        _e( 'Showing the single result', 'car-repair-services' );
    } elseif ( $total <= $per_page || -1 === $per_page ) {
printf( _n( 'Showing all %d result', 'Showing all %d results', $total, 'car-repair-services'
), $total );

```

```

    } else {
$first = ( $per_page * $current ) - $per_page + 1;
$last  = min( $total, $per_page * $current );
printf( _nx( 'Showing the single result', 'Showing %1$d&ndash;%2$d of %3$d results', $total,
'with first and last result', 'car-repair-services' ), $first, $last, $total );
    }
?>
</span>
<?php
}

```

```

<?php

defined( 'ABSPATH' ) || exit;

get_header( 'shop' );

do_action( 'woocommerce_before_main_content' );

?>
<header class="woocommerce-products-header">
<?php if ( apply_filters( 'woocommerce_show_page_title', true ) ) : ?>
<h1 class="woocommerce-products-header__title page-title"><?php woocommerce_page_title();
?></h1>
<?php endif; ?>

<?php
do_action( 'woocommerce_archive_description' );
?>
</header>
<?php
if ( woocommerce_product_loop() ) {

do_action( 'woocommerce_before_shop_loop' );

    woocommerce_product_loop_start();

if ( wc_get_loop_prop( 'total' ) ) {
while ( have_posts() ) {
    the_post();
do_action( 'woocommerce_shop_loop' );

```

```

        wc_get_template_part( 'content', 'product' );
    }
}

woocommerce_product_loop_end();

do_action( 'woocommerce_after_shop_loop' );
} else {

do_action( 'woocommerce_no_products_found' );
}

do_action( 'woocommerce_after_main_content' );

do_action( 'woocommerce_sidebar' );

get_footer( 'shop' );

<?php

// If this file is called directly, abort.
if ( ! defined( 'WPINC' ) ) {
    wp_die();
}

if( ! class_exists( 'FlrtFilter' ) ):

class FlrtFilter{

public function init()
    {
global $flrt_sets, $wpc_not_fired;

$wpc_not_fired = true;
$flrt_sets = [];

$this->define( 'FLRT_PLUGIN_DIR', plugin_dir_path( __FILE__ ) );
$this->define( 'FLRT_PLUGIN_URL', plugin_dir_url( __FILE__ ) );
$this->define( 'FLRT_PLUGIN_BASENAME', plugin_basename( __FILE__ ) );
$this->define( 'FLRT_PLUGIN_VER', '1.4.5' );
$this->define( 'FLRT_PLUGIN_LINK', 'https://filtereverything.pro' );
$this->define( 'FLRT_PLUGIN_DEBUG', false );

```

```

$this->define( 'FLRT_TEMPLATES_DIR_NAME', 'filters' );

$this->define( 'FLRT_FILTERS_SET_POST_TYPE', 'filter-set' );
$this->define( 'FLRT_FILTERS_POST_TYPE', 'filter-field' );
$this->define( 'FLRT_PREFIX_SEPARATOR', '-' );
$this->define( 'FLRT_QUERY_TERMS_SEPARATOR', ';' );
$this->define( 'FLRT_STATUS_COOKIE_NAME', 'wpcContainersStatus' );
$this->define( 'FLRT_HIERARCHY_LIST_COOKIE_NAME', 'wpcHierarchyListStatus' );
$this->define( 'FLRT_OPEN_CLOSE_BUTTON_COOKIE_NAME', 'wpcWidgetStatus' );
$this->define( 'FLRT_TRANSIENT_PERIOD_HOURS', 12 );

```

```

require_once FLRT_PLUGIN_DIR . 'src/wpc-helpers.php';

```

```

    flrt_include( 'src/wpc-compat.php' );
    flrt_include( 'src/wpc-default-hooks.php' );
    flrt_include( 'src/wpc-third-party.php' );

    flrt_include( 'src/Plugin.php' );
    flrt_include( 'src/PostTypes.php' );
    flrt_include( 'src/Settings/TabInterface.php' );
    flrt_include( 'src/Settings/BaseSettings.php' );
    flrt_include( 'src/Settings/TabRenderer.php' );
    flrt_include( 'src/Settings/Container.php' );

    flrt_include( 'src/Entities/Entity.php' );

    flrt_include( 'src/Entities/TaxonomyEntity.php' );
    flrt_include( 'src/Entities/PostMetaEntity.php' );
    flrt_include( 'src/Entities/PostMetaNumEntity.php' );
    flrt_include( 'src/Entities/AuthorEntity.php' );

```

```

// Include PRO

```

```

flrt_include( 'pro/filters-pro.php' );

    flrt_include( 'src/Entities/DefaultEntity.php' );
    flrt_include( 'src/Entities/EntityManager.php' );

    flrt_include( 'src/Settings/Tabs/SettingsTab.php' );
    flrt_include( 'src/Settings/Tabs/PermalinksTab.php' );
    flrt_include( 'src/Settings/Tabs/ExperimentalTab.php' );
    flrt_include( 'src/Settings/Tabs/AboutProTab.php' );

    flrt_include( 'src/Settings/Filter.php' );

```

```

flrt_include('src/RequestParser.php');
flrt_include('src/UrlManager.php');
flrt_include('src/Chips.php');
flrt_include('src/Sorting.php');

flrt_include('src/Walkers/WalkerCheckbox.php');

flrt_include('src/TemplateManager.php');
flrt_include('src/WpManager.php');

flrt_include('src/Admin/FilterSet.php');
flrt_include('src/Admin/FilterFields.php');
flrt_include('src/Admin/Admin.php');
flrt_include('src/Admin/AdminHooks.php');
flrt_include('src/Admin/MetaBoxes.php');
flrt_include('src/Admin/Widgets/FilterWidget.php');
flrt_include('src/Admin/Widgets/ChipsWidget.php');
flrt_include('src/Admin/Widgets/SortingWidget.php');
flrt_include('src/Admin/Widgets.php');
flrt_include('src/Admin/Shortcodes.php');
flrt_include('src/Admin/Validator.php');

flrt_include('src/FormFields/Input.php');
flrt_include('src/wpc-api.php');

$this->registerHooks();

if( flrt_get_experimental_option( 'disable_woo_orderby' ) === 'on' ) {
if( ! function_exists('woocommerce_catalog_ordering') ){
function woocommerce_catalog_ordering()
    {
return false;
    }
}
}

public function registerHooks()
    {
// Convert old post_name format to new. Since v1.1.24
add_action( 'init', [ $this, 'convertSetLocations' ], -1 );

// Backward compatibility. From v1.3.2
add_action( 'init', [ $this, 'convertShowChipsInContent' ], -1 );

```

```

        add_action( 'init', [ $this, 'oneTwoThreeGo' ] );

        add_action( 'init', [ $this, 'loadTextdomain' ], 0 );

        register_activation_hook(__FILE__, [ 'FilterEverything\Filter\Plugin',
'activate' ] );

        register_uninstall_hook(__FILE__, [ 'FilterEverything\Filter\Plugin',
'uninstall' ] );

        add_action( 'after_switch_theme', [ 'FilterEverything\Filter\Plugin', 'switchTheme' ]
);
    }

    public function convertShowChipsInContent()
    {
        // Backward compatibility. From v1.3.2
        $filter_settings = get_option( 'wpc_filter_settings' );

        if ( isset( $filter_settings[ 'show_terms_in_content' ] )
&& $filter_settings[ 'show_terms_in_content' ] === 'on' ) {
            $new_chips_hooks = [];
            $theme_dependencies = flrt_get_theme_dependencies();

            if ( flrt_is_woocommerce() ) {
                $new_chips_hooks[] = 'woocommerce_no_products_found';
                $new_chips_hooks[] = 'woocommerce_archive_description';
            }

            if ( isset( $theme_dependencies[ 'chips_hook' ] ) && is_array( $theme_dependencies[ 'chips_hook' ] ) ) {
                foreach ( $theme_dependencies[ 'chips_hook' ] as $compat_chips_hook ) {
                    $new_chips_hooks[] = $compat_chips_hook;
                }
            }

            $filter_settings[ 'show_terms_in_content' ] = $new_chips_hooks;
            update_option( 'wpc_filter_settings', $filter_settings );
        }
    }

    public function convertSetLocations()
    {
        if( is_admin() ) {

            global $wpdb;

```

```

// Convert separator from ":" to "__" and from -1 to 1
$sql = [];
$sql[] = "SELECT {$wpdb->posts}.ID, {$wpdb->posts}.post_name";
$sql[] = "FROM {$wpdb->posts}";
$sql[] = "WHERE {$wpdb->posts}.post_type = '%s'";
$sql[] = "AND {$wpdb->posts}.post_name REGEXP '[\:\:]+'";
$sql[] = "OR {$wpdb->posts}.post_name = '-1'";

$sql = implode(" ", $sql);
$sql = $wpdb->prepare($sql, FLRT_FILTERS_SET_POST_TYPE);

$results = $wpdb->get_results($sql, ARRAY_A);

if (!empty($results)) {

    foreach ($results as $row) {
        $update = [];

        if (!isset($row['post_name']) || !isset($row['ID'])) {
            continue;
        }

        if( $row['post_name'] == '-1' ){
            $new_post_name = '1';
        }else{
            $new_post_name = str_replace(":", "__", $row['post_name']);
        }

        $update[] = "UPDATE {$wpdb->posts}";
        $update[] = "SET {$wpdb->posts}.post_name = '%s'";
        $update[] = "WHERE {$wpdb->posts}.ID = %s";

        $updateSql = implode(" ", $update);

        $updateSql = $wpdb->prepare($updateSql, $new_post_name, $row['ID']);

        $wpdb->query($updateSql);
    }
}
}

```

```
public function loadTextdomain()
{
    load_plugin_textdomain( 'filter-everything', false, dirname(FLRT_PLUGIN_BASENAME)
. '/lang' );
}

public function oneTwoThreeGo()
{
new \FilterEverything\Filter\Plugin();
}

public function define( $name, $value = true )
{
if( ! defined( $name ) ) {
    define( $name, $value );
}
}

}

function flrt_filter()
{
global $wpcFilter;

if( ! isset( $wpcFilter ) ) {
$wpcFilter = new FlrtFilter();
$wpcFilter->init();
}

return $wpcFilter;

}

flrt_filter();

endif;
```