

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

Розроблення інформаційної системи обліку замовлень в СТО з асоціативним
пошуком

Виконав: студент 2 курсу, групи 601-ТН
спеціальності

122 Комп'ютерні науки

(шифр і назва спеціальності)

Бойко В.В.

(прізвище та ініціали)

Керівник Бесєдін В.Ф.

(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

**«Розроблення інформаційної системи обліку замовлень в СТО з
асоціативним пошуком»**

Студента групи 601-ТН Бойко Владислава Вячеславовича

Керівник роботи
доктор економічних наук,
професор Бесєдін В.Ф.

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

Полтава – 2021

РЕФЕРАТ

Пояснювальна записка містить: 72 с., 40 рис., 21 джерел, 1 додаток.

Об'єкт дослідження – діяльність та web-сайти підприємства з системою асоціативного пошуку інформації

Предмет досліджень – інструментальні засоби розробки інформаційних систем, web-сайтів.

Мета кваліфікаційної роботи – розроблення і впровадження інформаційної системи підприємства з асоціативним пошуком інформації.

Методи – технології об'єктно-орієнтованого програмування та проектування, уніфікована мова моделювання UML, теорія баз даних, системний аналіз, технології веб-програмування.

Ключові слова: база даних, пошук інформації, інформаційна система, Yii Framework, CSS, APACHE PHP, JAVASCRIPT, MySQL, JQUERY, HTML, сайт, web-сервер.

ABSTRACT

The explanatory note contains: 72 p., 40 pic., 21 sources, 1 appendix.

The object of research is the activity and web-sites of the enterprise with the system of associative information search

The subject of research - tools for developing information systems, web-sites.

The purpose of the qualification work is the development and implementation of the information system of the enterprise with associative search of information.

Methods - object-oriented programming and design technologies, unified UML modeling language, database theory, systems analysis, web programming technologies.

Keywords: database, information retrieval, information system, Yii Framework, CSS, APACHE PHP, JAVASCRIPT, MySQL, JQUERY, HTM, site, web server.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ	6
ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ	9
1.1 Актуальність роботи	9
1.2 Поняття інформаційної системи	10
1.2.2 Фактори, що обумовлюють впровадження систем.	12
1.2.3 Сучасний стан розвитку інформаційних систем.	13
1.2.4 Аналіз причин неефективної роботи інформаційних систем. .	15
1.3 Постановка задачі	18
РОЗДІЛ 2 ТЕОРЕТИЧНА ЧАСТИНА СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.	21
2.1. Діяльність "AJZAA"	22
2.2. Розробка програмного забезпечення	24
2.2.1 Контекстна діаграма	24
2.2.2 Діаграма прецедентів	25
2.2.3 Діаграма діяльності	26
2.3 Розробка ER–моделі предметної області	28
2.4 Структура інформаційної системи	29
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА	31
3.1 Аналіз та вибір мови програмування для розробки ІС.	31
3.1.1 Вибір клієнтської мови програмування.	31
3.1.2 Вибір серверної мови програмування.	32
3.1.4 Вибір локального сервера.	33
3.1.5 Вибір візуального середовища програмування.	34

	5
3.2 Вибір сервера баз даних для ІС.....	36
3.3 Схема даних інформаційної системи.....	37
3.4 Проектування дизайну сторінок сайту	41
РОЗДІЛ 4 ТЕСТУВАННЯ.....	54
4.1 Тестування ІС	54
4.2 Тест-план	54
ВИСНОВКИ	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
ДОДАТОК А КОД ПРОГРАМИ.....	61

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

AIC – Автоматизована інформаційна система

СУБД – Система управління базами даних

ПЗ – Програмне забезпечення

PHP – це мова програмування, спеціально розроблена для написання web-додатків (сценаріїв), що виконуються на Web-сервері.

HTML – мова розмітки гіпертексту.

IC - Інформаційна система .

MySQL- вільна система керування реляційними базами даних.

IC – інформаційна система.

OC – операційна система.

ПП – програмний продукт.

ЗМІ - засоби масової інформації.

ПІС - інформаційно-пошукова система.

ООП – об’єктно-орієнтоване програмування.

XML – extensible markup language – розширювана мова розмітки.

DHTML – dynamic HTML.

CSS – cascading style sheets – каскадні таблиці стилів.

CMS – це програмне забезпечення, призначене для конструювання і керування веб-сайтами.

Yii – yes it is – це високопродуктивний веб-фреймворк, написаний на PHP, реалізує парадигму модель-вид-контролер.

HTTP – hyper text transfer protocol – протокол передавання гіпертексту.

ООБД – об’єктно-орієнтована база даних.

ВСТУП

В XXI столітті часто використовуються інформаційні системи та Інтернет. В наш час вся потрібна інформація знаходиться в Інтернеті, тому учбові заклади, компанії і тому подібне, створюють інформаційні системи для того, щоб користувачам було зручно знайти ту інформацію яка їм цікава та потрібна.

Інтернет - всесвітня система взаємополучених комп'ютерних мереж, що базуються на комплекті Інтернет-протоколів. Інтернет також називають мережею мереж. Інтернет складається з мільйонів локальних і глобальних приватних, публічних, академічних, ділових і урядових мереж, пов'язаних між собою з використанням різноманітних дротових, оптичних і бездротових технологій. Інтернет становить фізичну основу для розміщення величезної кількості інформаційних ресурсів і послуг, таких як взаємопов'язані гіпертекстові документи Всесвітньої павутини (World Wide Web – WWW) та електронна пошта.

В повсякденній мові слово Інтернет найчастіше вживається в значенні Всесвітнього павутиння і доступної в ньому інформації, а не у значенні самої фізичної мережі. Також вживаються терміни Всесвітня мережа, Глобальна мережа чи навіть одне слово Мережа, Інет, Тенета, Міжмережжя, Інтернетрі [1].

Веб-технологія повністю перевернула наші уявлення про роботу з інформацією та з комп'ютером взагалі. Виявилось, що традиційні параметри розвитку обчислювальної техніки – продуктивність, пропускна спроможність, ємкість пристроїв, що запам'ятовують, не враховували головного «вузького місця» системи – інтерфейсу з людиною. Застарілий механізм взаємодії людини з інформаційною системою стримував впровадження нових технологій і зменшував вигоду від їх застосування. І лише коли інтерфейс між людиною і комп'ютером був спрощений до природності сприйняття звичайною людиною, тільки тоді зріс інтерес до можливостей обчислювальної техніки [6].

Інформаційна система, як система управління, тісно пов'язується, як з системами збереження та видачі інформації, так із системами, що забезпечують обмін інформацією в процесі управління. Вона охоплює сукупність засобів та методів, що дозволяють користувачу збирати, зберігати, передавати і обробляти відібрану інформацію. Інформаційні системи існують з моменту появи суспільства, оскільки на кожній стадії його розвитку існує потреба в управлінні.

Інформаційна система з точки зору її змісту – сукупність соціальних, економічних, виробничих та інших даних, що відображають внутрішній і зовнішній стан об'єкта управління. Вона є невід'ємним елементом будь-якої системи управління незалежно від її рівня.

Завданням інформаційної системи є виробництво інформації, потрібної для ефективного управління всіма її ресурсами, створення інформаційного та технічного середовища для управління її діяльністю. Інформаційна система може існувати і без застосування комп'ютерної техніки – це питання економічної необхідності [6].

РОЗДІЛ 1

АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Актуальність роботи

Для виконання дипломної роботи була поставлена задача створити інформаційну систему підприємства (на базі підприємства AJZAA). Інформаційна система, сайт представляє організацію, називається корпоративним або офіційним сайтом організації. До офіційного сайту пред'являються більш жорсткі вимоги в частині інформаційного вмісту, графічного дизайну, навігації, хостингу.

Офіційний сайт має наступні розділи:

- 1) Новинна інформація.
- 2) Нормативні документи, положення.
- 3) Напрямки діяльності.
- 4) Структура послуг.
- 5) Контактна інформація (список відповідальних осіб, їх посади, координати).

Основними факторами, які впливають на впровадження інформаційних систем, є потреби організацій та користувачів, а також наявність відповідних засобів для їх формування.

Причини, що спонукають організації впроваджувати інформаційні системи, з одного боку обумовлюються прагненням збільшити продуктивність повсякденних робіт чи усунути їх повторне проведення, а з іншого боку бажанням підвищити ефективність управління діяльністю організації за рахунок прийняття оптимальних та раціональних управлінських рішень.

Перша причина доволі прозора і для її реалізації достатньо впроваджувати стандартизовані системи обробки інформації. Успішне функціонування організації у значній мірі залежить від вдалого керівництва, яке базується на обґрунтуванні перспективних концепцій розвитку згідно з

своєчасною, достовірною та повною інформацією, яку може поставляти відповідна інформаційна система. Основне завдання інформаційної системи управління полягає у підпорядкуванні всіх внутрішніх процесів головним цілям організації. Для цього необхідно скоординувати процеси, пов'язані з діяльністю організації таким чином, щоб вони максимально забезпечували виконання поставлених задач в єдиному інформаційному полі. Тільки таким чином інформаційна озброєність організації починає безпосередньо впливати на ефективність її діяльності [2].

До основних напрямків автоматизації інформаційно-управлінської діяльності в організаційних структурах відносять:

- автоматизацію обробки документів шляхом впровадження систем для обробки тексту, автоматизацію обміну інформацією через різноманітні види комунікацій (які включають АТС підприємства, відеотермінальні системи, локальну комп'ютерну мережу, телекопіювальні апарати, відеоінформаційні системи);
- автоматизацію діяльності менеджерів на базі комп'ютерних систем комплексних інформаційних систем, які надають допомогу в прийнятті рішень, та електронних секретарів, що дозволяє підвищити рівень організації праці менеджерів на якісно вищій щабель [7].

1.2 Поняття інформаційної системи

1.2.1 Визначення інформаційної системи. Інформаційна система – це сукупність організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів.

Інформаційні системи здавна знаходять (в тому чи іншому вигляді) досить широке застосування в життєдіяльності людства. Це пов'язано з тим, що для існування цивілізації необхідний обмін інформацією – передача знань, як між окремими членами і колективами суспільства, так і між різними поколіннями.

Інформаційні системи існують з моменту появи суспільства, оскільки на кожній стадії його розвитку існує потреба в управлінні. Місією інформаційної системи є переробка інформації, потрібної для ефективного управління всіма ресурсами організації, створення інформаційного та технічного середовища для управління її діяльністю [7].

Інформаційна система може існувати і без застосування комп'ютерної техніки – це питання економічної доцільності.

Інформаційні системи включають в себе: технічні засоби обробки даних, програмне забезпечення і відповідний персонал. Чотири складові частини утворюють внутрішню інформаційну основу:

- засоби фіксації і збору інформації;
- засоби передачі відповідних даних та повідомлень;
- засоби збереження інформації;
- засоби аналізу, обробки і представлення інформації.

В будь-якій інформаційній системі управління вирішуються задачі трьох типів:

- задачі оцінки ситуації (деколи їх називають задачами розпізнавання образів);
- задачі перетворення опису ситуації (розрахункові задачі, задачі моделювання);
- задачі прийняття рішень (в тому числі і оптимізаційні).

Найдавнішими і найпоширенішими ІС слід вважати бібліотеки. І, дійсно, здавна в бібліотеках збирають книжки (або їх аналоги), зберігають їх, дотримуючись певних правил, створюють каталоги різного призначення для полегшення доступу до книжкового фонду. Видаються спеціальні журнали та довідники, що інформують про нові надходження, ведеться облік видачі.

Найстаріші (у моральному і у фізичному розумінні) ІС повністю базувалися на ручній праці. Пізніше їм на зміну прийшли різні механічні пристрої для обробки даних (наприклад, для сортування, копіювання, асоціативного пошуку тощо). Наступним кроком стало впровадження

автоматизованих інформаційних систем (АІС), тобто систем, де для забезпечення інформаційних потреб користувачів використовується ЕОМ зі своїми носіями інформації. В наш час – епоху інформаційної революції – розробляється і впроваджується велика кількість найрізноманітніших АІС з дуже широким спектром використання [7].

1.2.2 Фактори, що обумовлюють впровадження інформаційних систем. Основними факторами, які впливають на впровадження інформаційних систем, є потреби організацій та користувачів, а також наявність відповідних засобів для їх формування. Найсуттєвіше на розвиток інформаційних систем вплинули досягнення в галузі комп'ютерної техніки та телекомунікаційних мереж.

Причини, що спонукають організації впроваджувати інформаційні системи, з одного боку обумовлюються прагненням збільшити продуктивність повсякденних робіт чи усунути їх повторне проведення, а з іншого боку бажанням підвищити ефективність управління діяльністю організації за рахунок прийняття оптимальних та раціональних управлінських рішень. Перша причина доволі прозора і для її реалізації достатньо впроваджувати стандартизовані системи обробки інформації. Успішне функціонування організації у значній мірі залежить від вдалого керівництва, яке базується на обґрунтуванні перспективних концепцій розвитку згідно з своєчасною, достовірною та повною інформацією, яку може поставляти відповідна інформаційна система. Основне завдання інформаційної системи управління полягає у підпорядкуванні всіх внутрішніх процесів головним цілям організації. Для цього необхідно скоординувати процеси, пов'язані з діяльністю організації таким чином, щоб вони максимально забезпечували виконання поставлених задач в єдиному інформаційному полі. Тільки таким чином інформаційна озброєність організації починає безпосередньо впливати на ефективність її діяльності [8].

До основних напрямків автоматизації інформаційно-управлінської діяльності в організаційних структурах відносять:

- автоматизацію обробки документів шляхом впровадження систем для обробки тексту, автоматизацію обміну інформацією через різноманітні види комунікацій (які включають АТС підприємства, відеотермінальні системи, локальну комп'ютерну мережу, телекопіювальні апарати, відеоінформаційні системи);
- автоматизацію діяльності менеджерів на базі комп'ютерних систем комплексних інформаційних систем, які надають допомогу в прийнятті рішень, та електронних секретарів, що дозволяє підвищити рівень організації праці менеджерів на якісно вищій щабель.

Впровадження інформаційних систем дозволяє менеджеру отримувати оперативний доступ до довільної нагромадженої інформації з тим, щоб в подальшому ефективно її використовувати для вирішення поставлених задач [10].

1.2.3 Сучасний стан розвитку інформаційних систем. Для сучасних умов характерне застосування високоефективних внутрішньофірмових систем інформації, що ґрунтуються на використанні найновіших інформаційних технологій, зокрема єдиної локальної комп'ютерної мережі. Управлінська внутрішня інформаційна система представляє собою сукупність інформаційних процесів для задоволення потреб в інформації на різних рівнях прийняття рішень. Інформаційна система включає компоненти обробки інформації, внутрішні і зовнішні канали передачі.

Інформація, особливо її автоматизована обробка, і тепер залишається важливим фактором підвищення ефективності діяльності будь-якої організації. Важливу роль у використанні інформації відіграють способи її реєстрації, обробки, нагромадження і передачі; систематизоване збереження інформації і її видача в потрібній формі; виробництво нової числової, графічної та іншої інформації [11].

В сучасних умовах у великих організаціях створені і ефективно діють інформаційні системи, які обслуговують процес підготовки і прийняття управлінських рішень і вирішують наступні задачі: обробку даних, обробку інформації, реалізацію інтелектуальної діяльності з метою створення інформації. Управлінські інформаційні системи послідовно реалізують принципи єдності виробничого процесу та інформаційного процесу супроводу через застосування технічних засобів збору, нагромадження, обробки і передачі інформації в поєднанні з використанням аналітичних методів математичної статистики і моделей прогнозно-аналітичних розрахунків та інших необхідних прикладних засобів. У виробничо-господарській структурі підприємства забезпечується узагальнення інформації “знизу - вгору”, конкретизація інформації “зверху - вниз”, а також уніфікується інформаційний процес, спрямований на отримання науково-технічної, планової, контрольної, облікової і аналітичної інформації [8].

Підвищення ефективності використання інформаційних систем досягається шляхом наскрізної структури і сумісності інформаційних систем, які дозволяють усунути дублювання і забезпечують багатократне використання інформації, встановлюють визначені інтеграційні зв'язки, обмежують кількість показників, зменшують обсяг інформаційних потоків, підвищують рівень використання інформації. Інформаційна система повинна підтримувати такі функції, як надання інформації (наприклад, потрібної користувачам для вирішення науково-виробничих задач) та створення найзручніших умов для її поширення (наприклад, проведення адміністративно-організаційних, науково-дослідних і виробничих заходів, які забезпечують її ефективне розповсюдження).

Сучасна інформаційна система в заданій сфері діяльності організації дозволяє забезпечити вирішення таких завдань:

- прямиий, своєчасний доступ до інформаційного продукту (точну інформацію про хід виробничого процесу в просторі та часі);

- ефективну координацію внутрішньої діяльності та оперативне розповсюдження різноманітних повідомлень;
- ефективнішу взаємодію із суміжниками по технологічних маршрутах за рахунок використання більш інформованих та наочних засобів відображення та передачі-прийому повідомлень;
- виділення необхідного і неперервного часу для менеджерів всіх ланок на такі високоефективні види діяльності, як аналіз та прийняття рішень за рахунок зменшення часу на здійснення малопродуктивної діяльності;
- використання якісно кращої технології системного аналізу та проектування оперативного управління на нижній та середніх ланках управління виробництвом [8].

1.2.4 Аналіз причин неефективної роботи інформаційних систем.

Нав'язлива політика владних структур на застосування інформаційних систем та подальша неефективність використання інформаційних систем призвели до несприйняття багатьма колишніми працівниками сфери менеджменту проблем впровадження інформаційних технологій. Це пояснюється такими особливостями колишніх розробок інформаційних систем:

- переважання інтуїтивної технології вибору і узагальнення управлінських рішень через обмеженість в системно-технічних принципах; роздільне функціонування основних систем інформаційного забезпечення в галузях виробничої, організаційної, маркетингової, фінансової, бухгалтерської, кадрової та проектно-конструкторської діяльності підприємства не забезпечували комплексного вирішення багатопланових проблем організації і не могли дати повний аналіз поточного стану організації;
- надмірно централізована обробка інформації шляхом концентрації інформаційних потоків в невеликій кількості пунктів, віддалених від безпосереднього користувача проміжними ланками технічного персоналу; домінування неавтоматизованих функцій інформаційних систем через технічну неможливість або економічну неефективність автоматизації даних функцій

управління не забезпечували прямого доступу працівників сфери менеджменту до інформації, а також штучно відчужували первинного давача інформації від самої інформації, створюючи перепони в часі та просторі;

- використання багаторівневої технології проектування, кожна стадія якої виконувалася спеціалістами різної спеціальності і кваліфікації; порівняно тривале проектування систем (3-5 роки) і планована довготривала експлуатація з мінімальними змінами (10-15 років) – штучно затримували еволюцію підприємства, оскільки затрати на пристосування інформаційних систем до нових умов використання могли перевищити витрати на розробку та впровадження [6].

Наведені вище особливості в час ринкових реформ мають негативний відтінок, і тому на сьогодні при розробці інформаційних систем насамперед потрібно мінімізувати негативний вплив цих факторів, що і дозволяють зробити сучасні інформаційні технології.

В наш час відсутність можливості взаємодії окремих засобів автоматизації чи навіть окрема технологія може стати стримуючим фактором, що робить використання інформаційних систем нераціональним. Більше того, для найконсервативнішої частини керівників використання інформаційних технологій стає приводом для відмови від відповідних капіталовкладень.

Розміри необхідного капіталу також можуть служити, як підтримуючою прогрес силою, так і гальмом для впровадження інформаційних систем. Не багато керівників будуть стверджувати, що значні інвестиції в автоматизацію організації, розраховані на довготермінову перспективу, є вирішальними в питаннях виживання. Для короткотермінової перспективи багато хто ставить під сумнів окупність інвестицій, оскільки не має чіткого уявлення про місце інтелектуальних систем в управлінні організацією [6].

1.3 Асоціативний пошук

Розробники алгоритму постаралися відтворити динамічну, асоціативну природу людського мислення. Вони виходили з того, що при прийнятті рішень люди, використовують для обробки інформації неієрархічні способи аналізу. Обдумуючи задачу, кожна людина йде до висновку своїм шляхом. Користувачам надається можливість аналізувати дані відповідно до власних розумових процесів.

Традиційні методи пошуку і фільтрації інформації були розроблені для бібліотечних баз даних, тобто для інформаційних ресурсів обмеженого обсягу і заздалегідь відомої структури. Створення глобальної мережі і вихід Web за рамки інтересів наукового співтовариства призвели до того, що число постачальників інформації стало стрімко рости, при тому, що опублікована ними інформація не мала однорідної структури. Наступний інформаційний вибух став викликом стандартним інформаційним технологіям. Нові масштаби з одного боку зробили аутсайдерами деякі раніше конкурентоспроможні інтелектуальні технології, а з іншого – стимулювали інтенсивні дослідження в області лінгвістичних та імовірнісних методів обробки текстової інформації та нових методів навігації в неоднорідному інформаційному морі [6].

Асоціативний пошук ґрунтується на двох ключових принципах:

- всі дані зберігаються в пам'яті комп'ютера (ОЗУ);
- обчислення виконуються в реальному часі.

Ці архітектурні рішення підкріплюються двома важливими тенденціями в розвитку комп'ютерів. Перша – перехід від 32- до 64-розрядних обчислень, що призвело до експоненціального збільшенню розміру ОЗУ комп'ютера. В даний час можна придбати сервери, де ОЗУ досягає 512 Гбайт, а не так давно, більшість серверів розташовувало ОЗУ ємністю 4 Гбайт. Завдяки збільшенню доступною пам'яті стало можливо перемістити сховище даних з диска безпосередньо в ОЗУ. Друга обставина – повсюдне поширення багатоядерних процесорів. В даний час широко застосовуються сервери, які мають 8

процесорів і в цілому 48 ядер. Зміни привели до серйозного збільшення обчислювальної потужності додатків, орієнтованих на паралельні обчислення. Високопродуктивні сервери пропонуються менш ніж за 50 000 дол. Зовсім недавно для досягнення такого рівня обчислювальних можливостей були потрібні спеціально сконструйовані комп'ютери вартістю сотні тисяч і навіть мільйони доларів [6].

Архітектура зберігання і обробки інформації в оперативній пам'яті (in-memory), дозволяє маніпулювати величезними обсягами даних, підтримуючи високий рівень інтерактивності. Стиснення даних при завантаженні в пам'ять дозволяє зберігати дані в ОЗУ ефективніше, ніж на диску, як це прийнято в традиційних реляційних базах даних. При передачі даних в пам'ять система будує карту посилань між елементами даних, щоб спростити візуалізацію їх зв'язків [6].

Завдяки здатності виконувати обчислення в реальному часі можна швидко розраховувати складні заходи і метрики. В новітніх системах закладена можливість розподіляти обчислення між всіма доступними ядрами процесорів і управляти робочим навантаженням одночасно для великої кількості користувачів. Крім того, платформа зберігає в кеш-пам'яті результати всіх запитів, тому типові обчислення виконуються мінімальна кількість разів.

1.4 Огляд існуючих в Інтернеті інформаційних систем про надання послуг автосервісу 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 автосервісу автомобілів:

1. «ROOLER» автосервіс автомобілів в місті Полтава представлений на рис. 1.1

Переваги:

- сучасний дизайн;
- гарна навігація;

Недоліки представленого сайту:

- недоліків на сайті не знайдено.

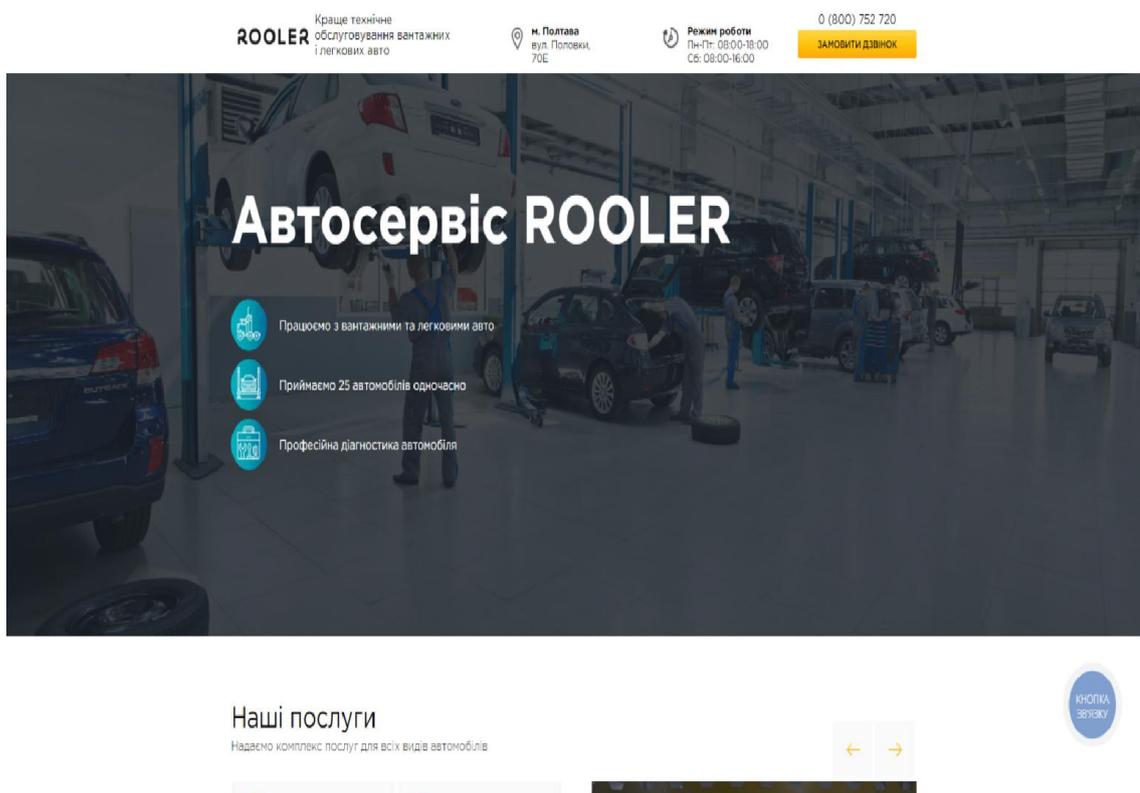


Рисунок 1.1 – Автосервіс автомобілів «ROOLER»

2. «OILER» Автосервіс автомобілів у Києві представлений на рис. 1.2

Переваги:

- зручна навігація;
- легкий та зручний інтерфейс;

Недоліки представленого сайту:

- застарілий дизайн;
- багато непотрібної інформації;

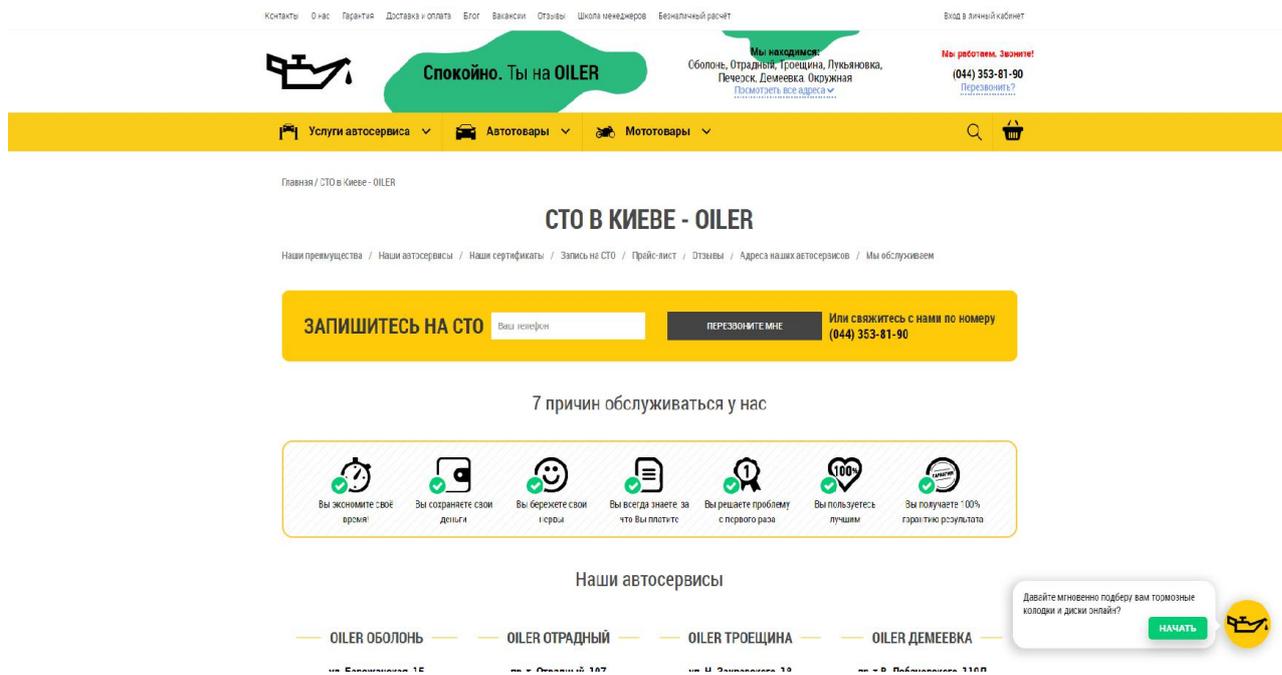


Рисунок 1.2 – «OILER» автосервіс автомобілів у Києві

3. «1A autoservice» автосервіс автомобілів у Києві представлений на рис. 1.3

Преваги:

- сучасний дизайн;
- зручна навігація;

Недоліки представленою сайту:

- незручна таблиця ціноутворення;
- багато непотрібної інформації.

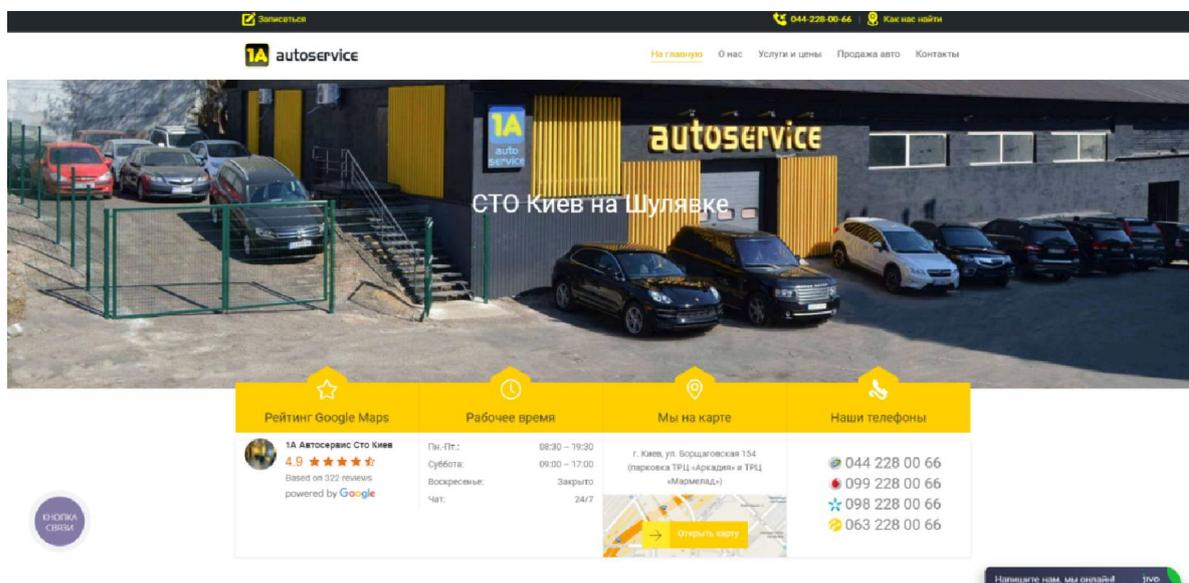


Рисунок 1.3 – «1A autoservice» автосервіс автомобілів у Києві

1.4 Постановка задачі

На основі проведеного аналізу була поставлена задача дослідження: спроектувати та програмно реалізувати Інформаційну систему підприємства автосервісу «AJZAA» з системою пошуку інформації (асоціативний пошук), яка повиненна забезпечувати виконання таких функцій:

- 1 Відображення інформації про підприємство на головній сторінці.
- 2 Реєстрація нових користувачів.
- 3 Користувачам:
 - 3.1 Переглядати інформацію.
 - 3.2 Здійснювати пошук необхідної інформації.
 - 3.3 Обмінюватись повідомленнями та файлами.
- 4 Адміністратору:
 - 4.1 Можливість редагувати дані, блокувати і видаляти користувачів.
 - 4.2 Редагування інформації.
 - 4.3 Управління існуючими ролями, створення і видалення ролей.
 - 4.4 Управління дозволами для ролей.

РОЗДІЛ 2

ТЕОРЕТИЧНА ЧАСТИНА СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Діяльність «AJZAA»

Компанія «AJZAA» готова запропонувати Вам широкий спектр послуг в сфері обслуговування та ремонту автомобілей та продажу запасних частин.

Додаткові відомості:

- усі види ремонтних робіт та послуг для імпортних та вітчизняних легкових автомобілів, малотоннажного транспорту комерційного призначення та причіпної техніки;
- виконуємо ремонт будь-якого масштабу та складності. У нас величезний вибір запчастин та розхідників для вашого автомобіля;
- гарантія на всі види робіт та послуг;
- можлива будь-яка зручна форма оплати.

Адреса: вул Автобазівська 5, м. Полтава Україна,

Телефон: 095-927-90-88

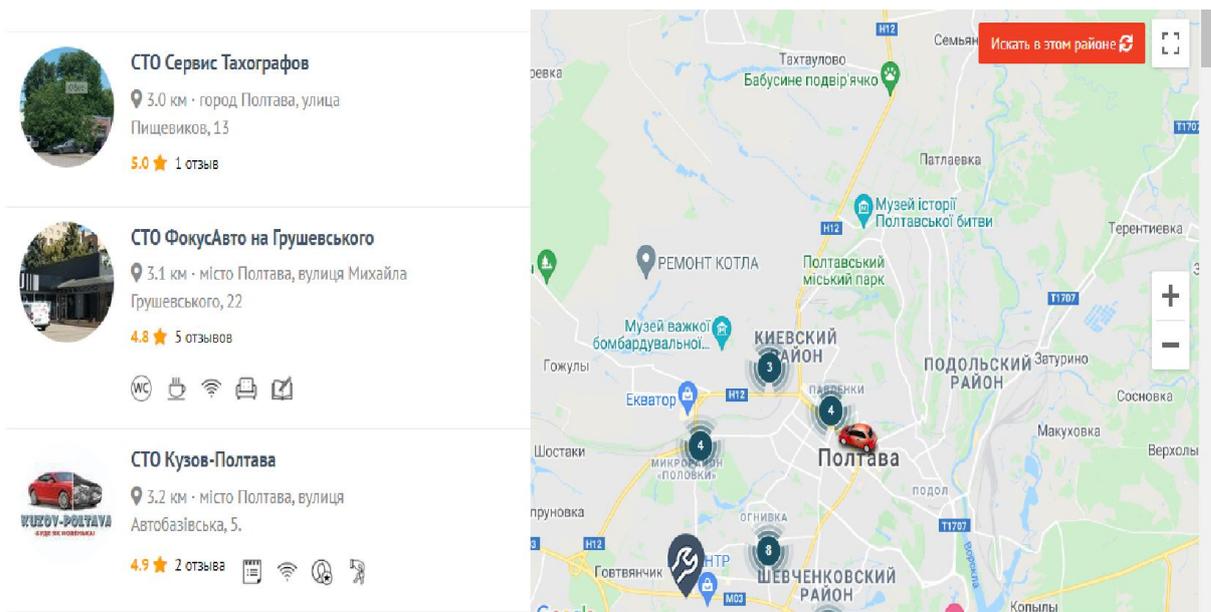


Рисунок 2.1 – Карта з СТО в Полтаві

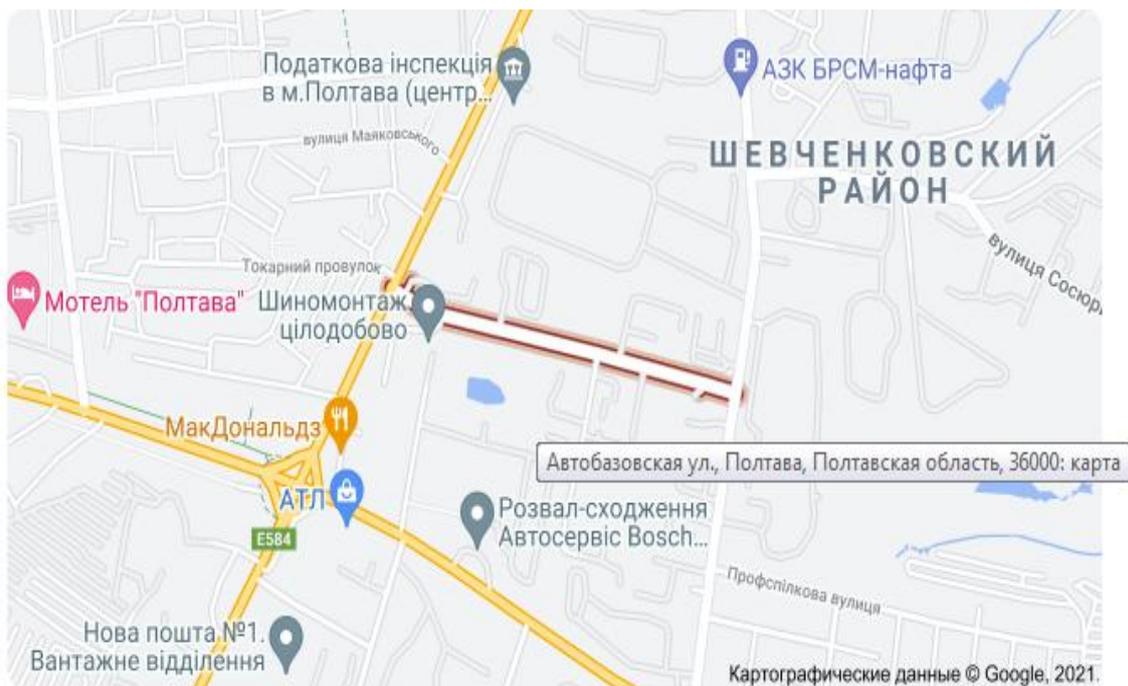


Рисунок 2.2 –Карта розміщення підприємства «AJZAA»



Рисунок 2.3 –Будівлі підприємства «AJZAA»

2.2 Розробка програмного забезпечення

Технологія створення інформаційної системи 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 [17].

2.2.1 Контекстна діаграма. Кожна модель повинна мати контекстну діаграму верхнього рівня, на якій об'єкт опису представлений поодиноким блоком, оточеним дугами. Дуги на цій діаграмі пов'язують об'єкт опису з навколишнім середовищем (зовнішніми об'єктами). 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 [14].

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 подфункція 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 [14].

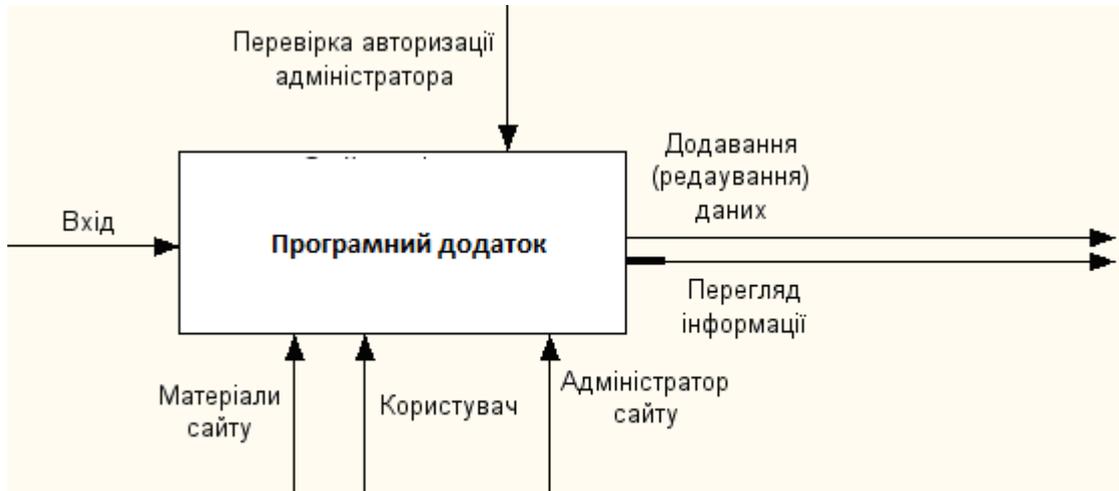


Рисунок 2.4 – Контекстна діаграма

2.2.2 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 [12].

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 [12].

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.

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

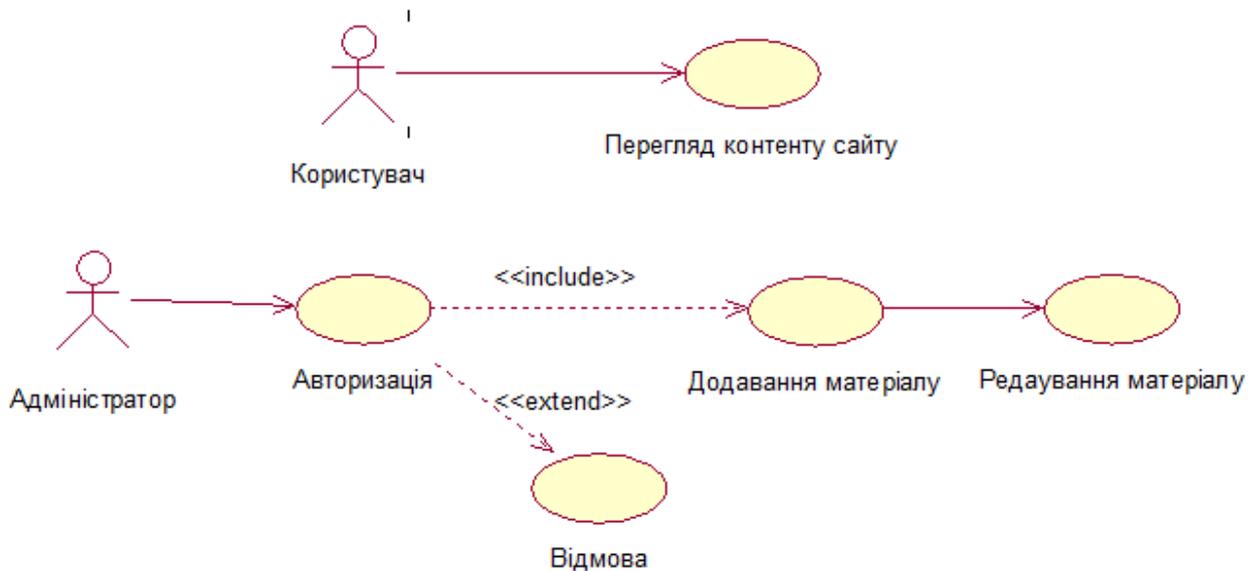


Рисунок 2.6 – Діаграма прецедентів

Дія (англ. action) є фундаментальною одиницею визначення поведінки в специфікації [13]. 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 [14].

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

- скруглені прямокутники позначають дії;
- ромби позначають рішення;
- риси 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.

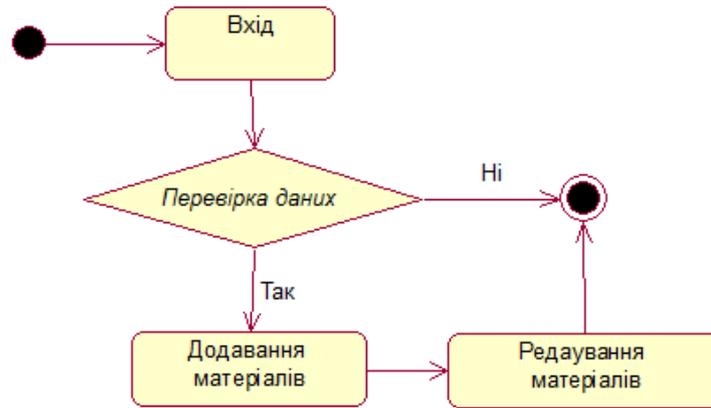


Рисунок 2.7– Діаграма діяльності

2.3 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 [17].

Кожний екземпляр сутності повинен відрізнятися 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 односточковий 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 [17].

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

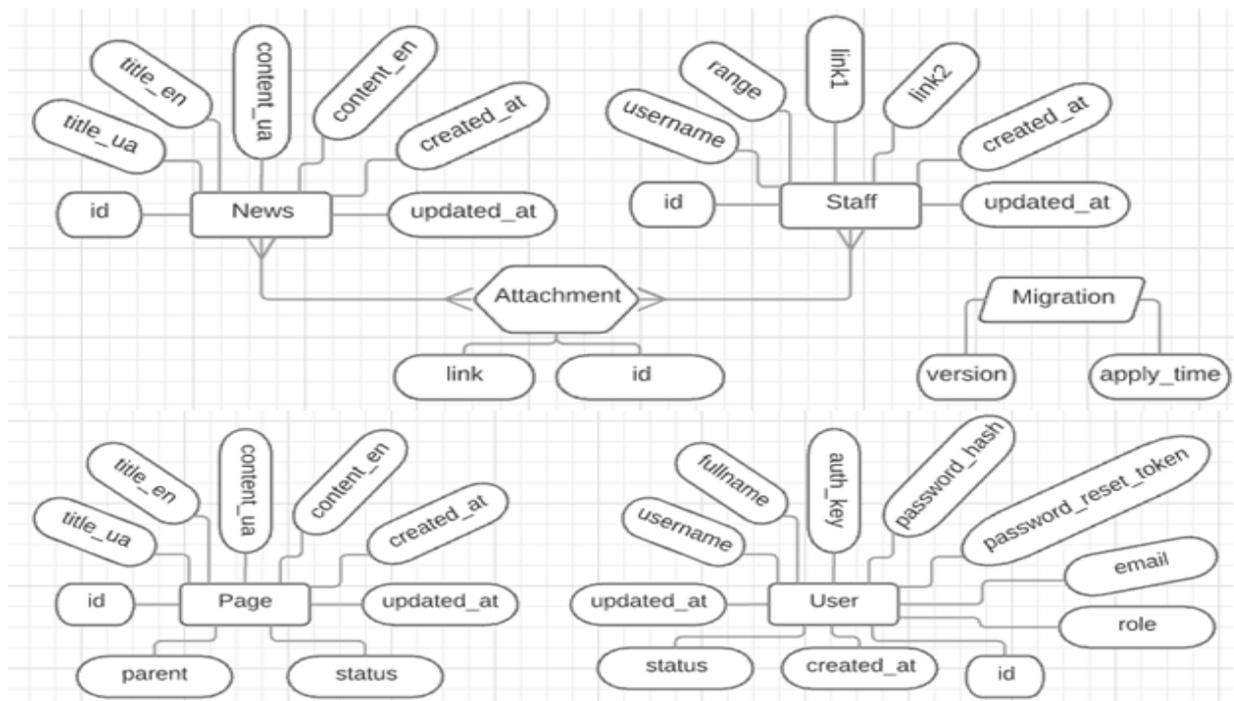


Рисунок 2.8– ER-модель

2.4 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. 2.8), оскільки вона є найдоцільніша в даному випадку.

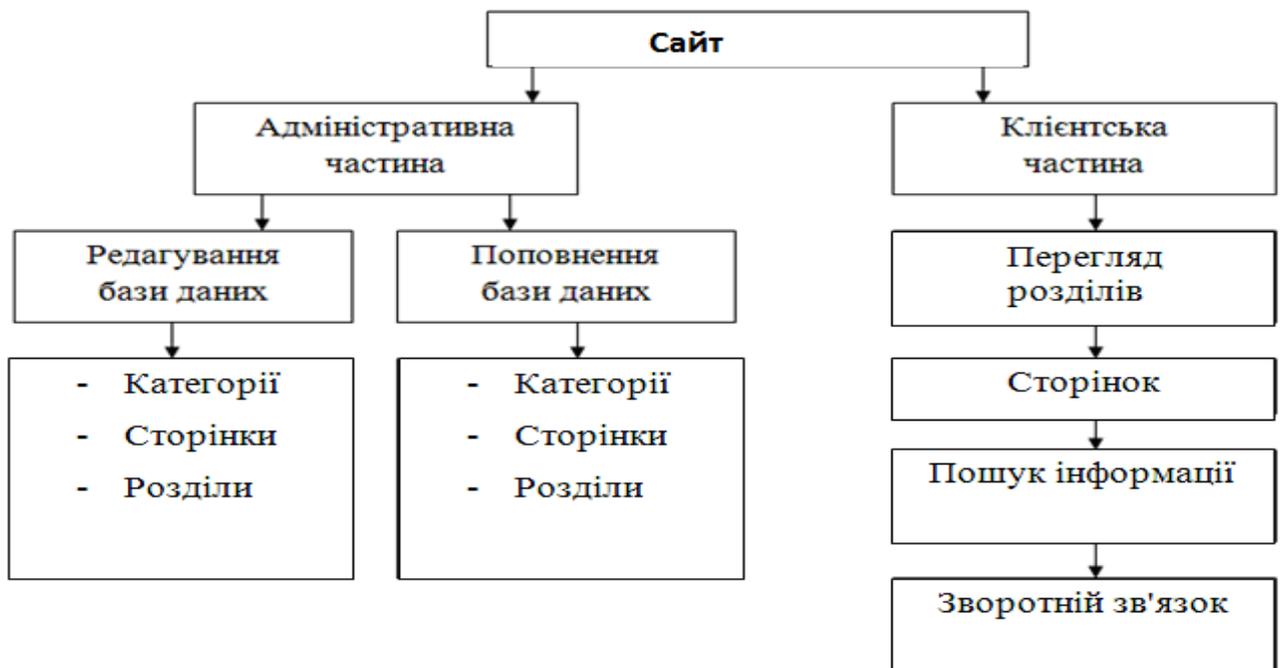


Рисунок 2.9 – Інформаційна структура

РОЗДІЛ 3

ПРАКТИЧНА ЧАСТИНА

3.1 Аналіз та вибір мови програмування для розробки ІС.

Існує кілька мов програмування розроблених спеціально для створення веб-сторінок та сайтів, зокрема PHP, Perl, Python, Ruby, ASP.NET, Java, Groovy. В свою чергу вони поділяються на клієнтські та серверні.

3.1.1 Вибір клієнтської мови програмування. Програми які написані на клієнтських мовах обробляються на стороні користувача, як правило, їх виконує браузер. Це і створює головну проблему клієнтських мов – результат виконання програми (скрипта) залежить від браузера користувача. Тобто, якщо користувач заборонив виконувати клієнтські програми, то вони виконуватися не будуть, як би не бажав цього програміст. Крім того, може статися таке, що в різних браузерах або в різних версіях одного і того ж браузера один і той же скрипт буде виконуватися по-різному. З іншого боку, якщо програміст покладає надії на серверні програми, то він може спростити їх роботу і знизити навантаження на сервер за рахунок програм, що виконуються на стороні клієнта, оскільки вони не завжди вимагають перезавантаження (генерацію) сторінки [17].

При виборі клієнтської мови доцільно використовувати HTML оскільки вона підтримується усіма відомими браузерами і працює майже безвідмовно. HTML (Мова розмітки гіпертекстових документів) – стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML обробляється браузером та відтворюється на екрані у зручному для людини вигляді [9].

3.1.2 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, 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.3.1). 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 [20].

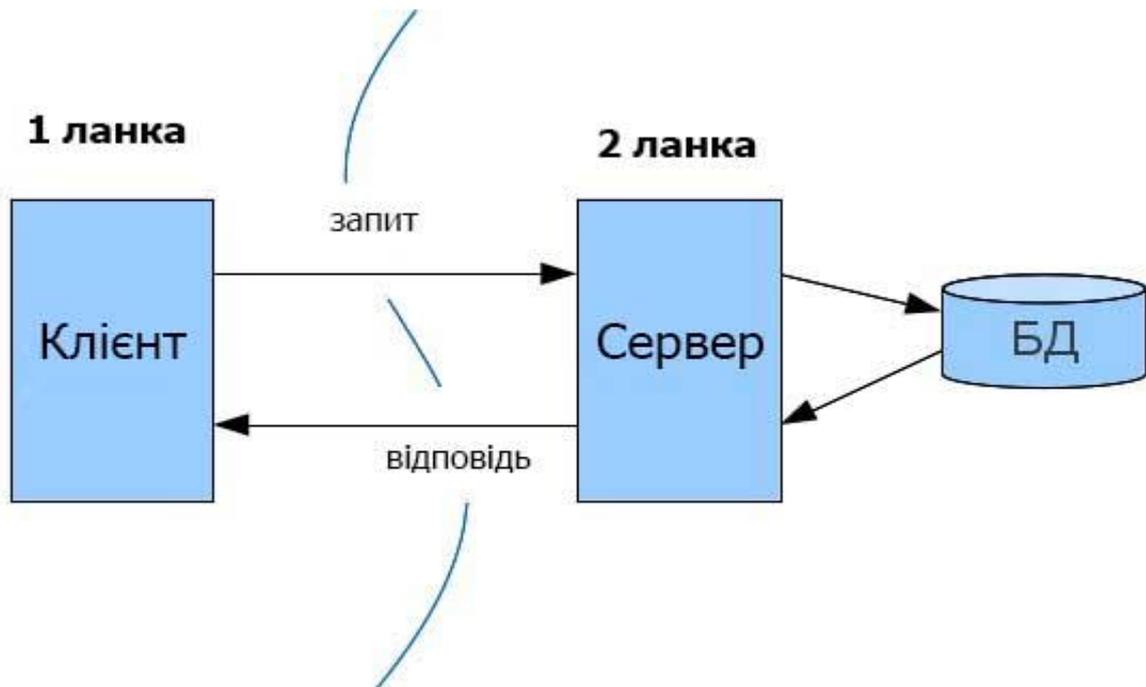


Рисунок 3.1 – Мережева організація

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-сервера. PHP є однією з найпоширеніших мов, що

використовуються у сфері веб-розробок. PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта [18].

3.1.3 Вибір середовища програмування. Середовище розробки – це комп'ютерна програма, що допомагає програмістові розробляти нове програмне забезпечення чи модифікувати (удосконалювати) вже існуюче.

Інтегровані середовища розробки зазвичай складаються з редактора серцевого коду, компілятора або інтерпретатора та засобів автоматизації збірки. Іноді сюди також входять системи контролю версій, засоби для профілювання, а також різноманітні засоби та утиліти для спрощення розробки графічного інтерфейсу користувача. Багато сучасних інтегрованих середовищ розробки також включають оглядач класів, інспектор об'єктів та діаграм ієрархії класів для використання об'єктно-орієнтованого підходу у розробці програмного забезпечення. Сучасні ІСР часто підтримують розробку на декількох мовах програмування [17].

Для зручної розробки веб-проекту знадобляться такі засоби програмування:

- локальний сервер/хостинг;
- візуальне середовище програмування;
- програма керування БД.

3.1.4 Вибір локального сервера. Після детального аналізу локальних серверів був вибраний LocalWP.

LocalWP – 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 [21].

3.1.5 Вибір візуального середовища програмування. Візуальне середовище програмування –це інтегроване середовище розробки програмних засобів, яке містить редактор вихідного коду, компілятор або інтерпретатор, засоби автоматизації збірки та 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 [17].

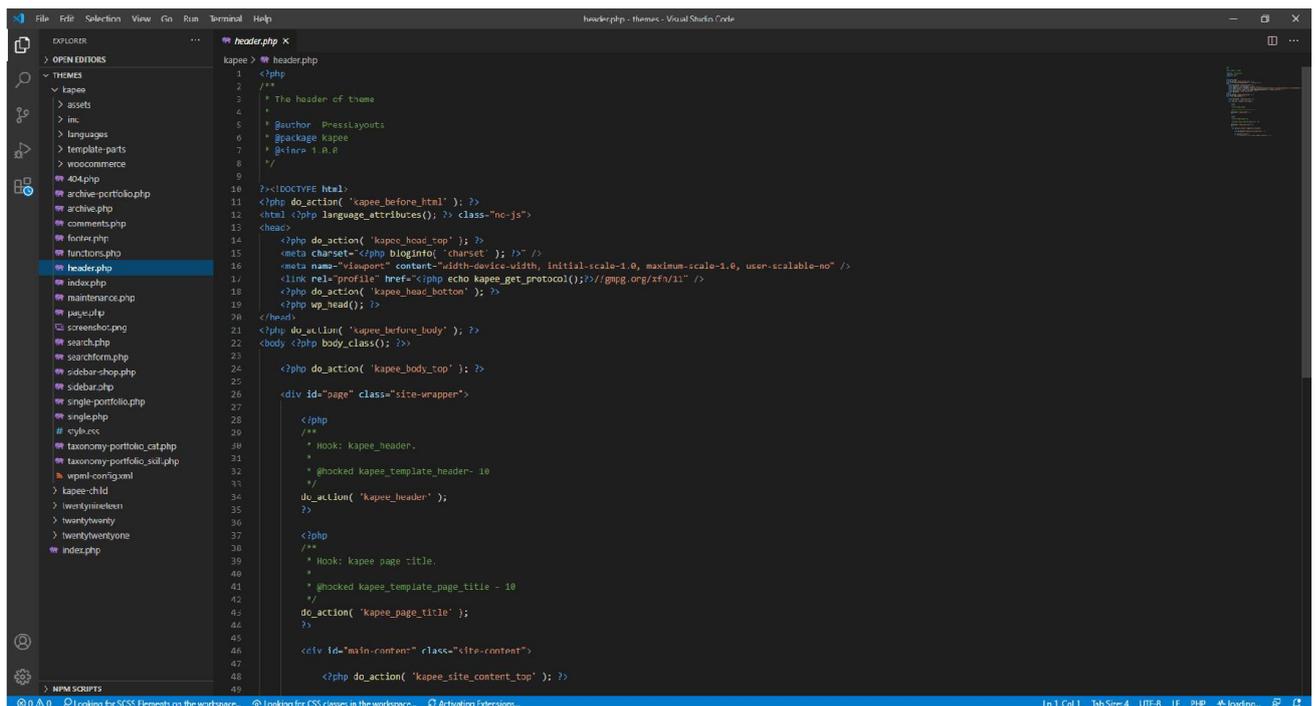


Рисунок 3.2 – Візуальне середовище програмування Visual Code.

Візуальним середовищем програмування був вибраний Visual Code (рис. 3.2). Visual Code – швидкий кросплатформенний текстовий редактор. Підтримує плагіни, розроблені за допомогою мови програмування C#.

Visual Code є відкритим програмним забезпеченням, але деякі його плагіни розповсюджуються з платною ліцензією, розробляються і підтримуються спільнотою розробників [6].

Насамперед необхідно зазначити, що LocalWP - це цілком і повністю портативний сервер. Ніяких системних сервісів, куп сміття в реєстрі і system32. можливо переносити його з собою на флешці (бажано на швидкісній), запускати на робочій або домашній машині без побоювання що щось не запрацює [18].

У разі відсутності на комп'ютері потрібних системних компонентів LocalWP необхідно встановити їх самостійно, 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 софта. Самою собою що за допомогою LocalWP можна запустити або зупинити сервер або відкрити потрібний домен. Набагато цікавіше буде дізнатися про специфічні «фічах», які роблять LocalWP особливим, дійсно особливим:

- докладний перегляд логів всіх компонентів в реальному часі;
- вибір HTTP, СУБД і PHP модулів в будь-якому поєднанні;
- підтримка SSL і кирилических доменів з коробки;
- підтримка алиасов або по іншому доменних покажчиків, а так само зручна форма їх налаштування;
- 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 LocalWP детально вивчаються і більшість з них реалізується [21].

3.2 Вибір сервера баз даних для ІС

Вибір системи управління баз даних (СУБД) являє собою складну багатопараметричну задачу і є одним з важливих етапів при розробці веб – додатків. Обраний програмний продукт повинен задовольняти як поточним, так і майбутнім потребам підприємства, при цьому слід враховувати фінансові витрати, розробку необхідного програмного забезпечення, а також навчання персоналу. Крім того, необхідно переконатися, що нова СУБД здатна принести підприємству реальні вигоди [19].

Очевидно, найбільш простий підхід при виборі СУБД заснований на оцінці того, якою мірою існуючі системи задовольняють основним вимогам створюваного проекту інформаційної системи. Більш складним і дорогим варіантом є створення випробувального проекту на основі кількох СУБД і подальший вибір найбільш гідного із кандидатів. Але і в цьому випадку необхідно обмежувати коло можливих систем, спираючись на певні критерії відбору. Взагалі, перелік вимог до СУБД, використовуваних при аналізі тієї чи іншої інформаційної системи, може змінюватися в залежності від поставлених цілей. Проте можна виділити кілька груп критеріїв:

- моделювання даних;
- особливості архітектури та функціональні можливості;
- контроль роботи системи;
- особливості розробки додатків;
- продуктивність;
- надійність;
- вимоги до робочого середовища;
- змішані критерії.

Навіть якщо просто відзначати наскільки гарні чи погані виділені параметри вразі кожної конкретної СУБД, то порівняння вже двох різних систем є трудомістким завданням. Тим не менш, чіткий і глибокий

порівняльний аналіз на підставі перерахованих вище критеріїв в будь – якому випадку допоможе раціонально вибрати відповідну систему для конкретного проекту, і витрачені зусилля не будуть марними. Перелік критеріїв допоможе усвідомити масштабність завдання і виконати її адекватну постановку [19].

MySQL – вільна система керування реляційними базами даних. Дана система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL – одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб – сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування [19].

MySQL – компактний багатопоточний сервер баз даних. Характеризується великою швидкістю, стійкістю і простотою використання. MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних.

Для некомерційного використання MySQL є безкоштовним.

- Переваги сервера MySQL:
 - простота у встановленні та використанні;
 - підтримується необмежена кількість користувачів, що одночасно працюють із базою даних;
 - кількість рядків у таблицях може досягати 50 млн;
 - висока швидкість виконання команд;
 - наявність простої і ефективної системи безпеки.
- Недоліки сервера MySQL:
 - відсутня підтримка зовнішніх (foreign) ключів;
 - відсутня підтримка тригерів і збережених процедур;

3.3 Схеми даних інформаційної системи

Створення власної інформаційної системи кафедри зв'язане з масою проблем:

По-перше, необхідно розробити даний проект.

По-друге, необхідно його просунути в пошукових системах, що займає, не мало часу, але принципово важливо, адже потрібно, щоб ви скрізь були на перших сторінках, інакше потенційні споживачі просто не зможуть вас знайти.

Спробуємо визначити основні елементи, які нам необхідні для створення сайту підприємства:

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

Велику увагу при створенні сайту потрібно виділяти розробці інтерфейсу. Інтерфейс – в широкому сенсі – визначена стандартами межа між взаємодіючими незалежними об'єктами. Інтерфейс задає параметри, процедури і характеристики взаємодії об'єктів [2].

Інтерфейс користувача – елементи і компоненти програми, які здатні робити вплив на взаємодію користувача з програмним забезпеченням. Зокрема:

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

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

Є декілька простих правил, що дозволяють інтерфейсу бути зрозумілим клієнтові.

Чим простіше, тим краще. Це зовсім не означає, що повинен бути тільки текст і контактна інформація, але на сайті не повинно бути даремної інформації, шрифт повинен легко читатися. Графічні елементи повинні бути були чіткими, виразними і швидко завантажуватися. На сайті неприпустимо використовувати анімацію і звук, які довго завантажуються і відволікають увагу покупця [2].

Людське око проглядає сторінки сайту зверху вниз. Найбільша увага зосереджується на верхній лівій частині сторінки. Тому у верхній частині сторінки сайту, як правило, розміщується найбільш важлива інформація: назва фірми, логотип, сама назва сайту і так далі. Чим простіше виконана верхня частина сторінки, тим легко запам'ятати назву сайту і саму фірму [7].

Типовою помилкою багатьох сайтів є нагромадження у верхній частині сторінки сайту авангардної і складної композиції з багатьох малюнків, серед яких ледве помітно розташовується назви фірми і назву сайту. В результаті людське око, скануючи дану частину сторінки сайту, не встигає розпізнати і запам'ятати потрібну інформацію. Відбувається перенасичення. Замість збільшення запам'ятованості, на таких сайтах, відбувається зниження запам'ятованості основних написів (малюнків) сайту, назви фірми і назва сайту [8].

Легкість пошуку потрібної інформації. Користувач повинен без зусиль знаходити інформацію, що цікавить його, і мати можливість отримати

Збільшенню відвідуваності сайту підприємства і кількості постійних користувачів сприяє також наявність зворотного зв'язку адміністратора з

користувачами. Важливо, щоб адміністратори щонайшвидше і максимально повно відповідали на листи.

Зворотною, невидимою користувачеві, стороною сайту є система управління. Вхід в систему адміністрування здійснюється тільки після введення адміністратором логіна і пароля. Адміністратор має можливість повністю управляти вмістом сайту підприємства:

- додавати або видаляти інформацію, описи і фотографії до них;
- редагувати розділи сайту підприємства;
- редагувати контактну інформацію;
- редагувати зміст заголовків і текстів листів, що відправляються користувачеві при реєстрації і відповіді на зворотній зв'язок;
- проглядати історію і статистику користувачів та інше.

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

Отже функціонуючу структуру сайту представимо наступним чином (Рис.3.5):



Рисунок 3.5 – Функціонуюча структура сайту

Враховуючи описані в розділі 1 та 2 загальні відомості про сайт, описані вище програмні засоби для реалізації сайту було розроблено інформаційна система підприємства «AJZAA» з використанням мови PHP та розроблена функціонуюча база даних інформації з використанням MySQL.

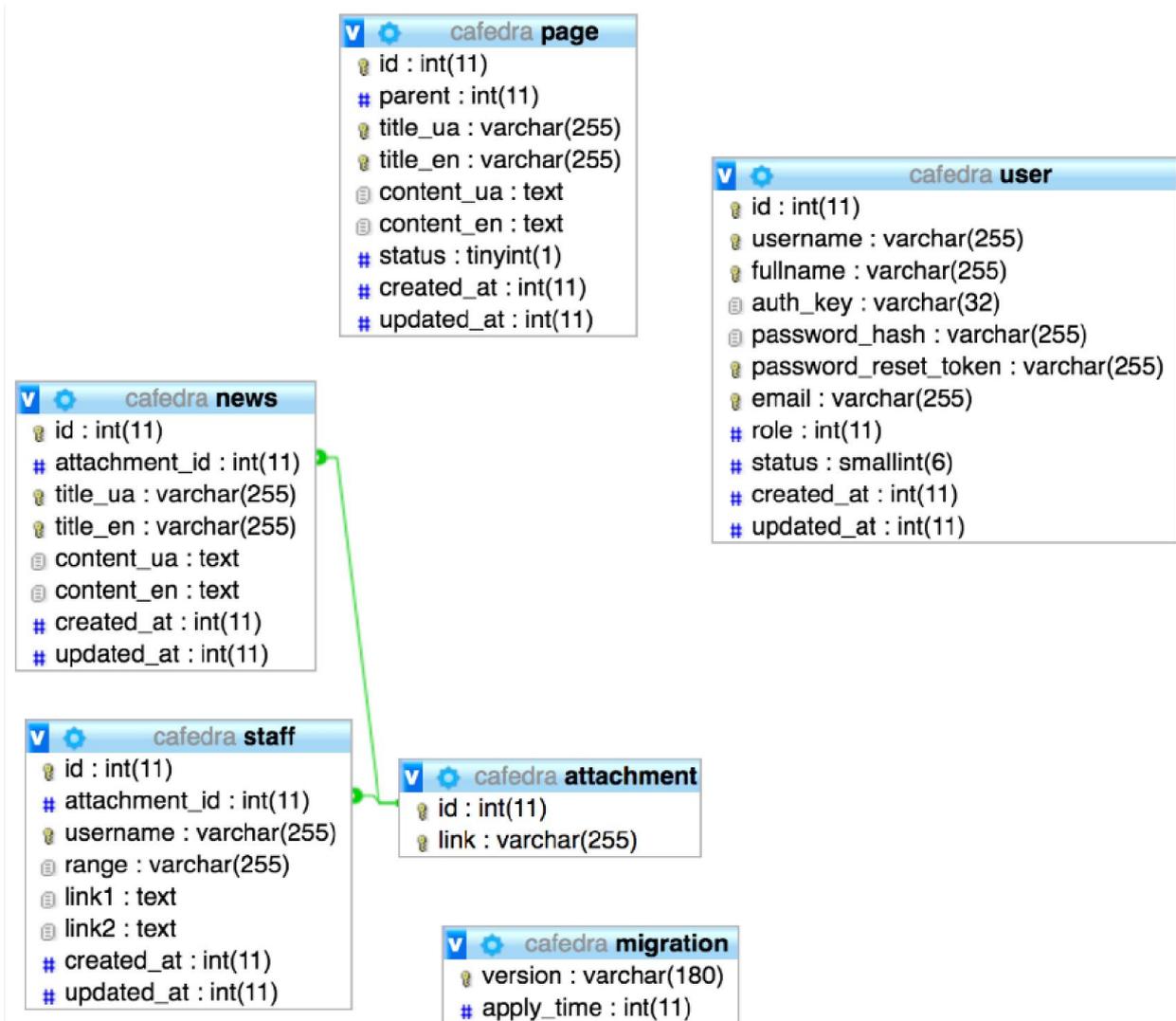


Рисунок 3.6 – Схема даних

3.4 Проектування дизайну сторінок сайту

Дизайн – галузь веб-розробки, що охоплює цілий ряд напрямів і дисциплін із створення та супроводу сайтів або веб-застосунків, таких як графічний веб-дизайн, проектування інтерфейсів, авторинг (у тому числі стандартизований код і власницьке програмне забезпечення), використання та оптимізація для пошукових систем. Найчастіше виконання різних етапів процесу веб-дизайну забезпечують відповідні спеціалісти (менеджер проекту, дизайнер, веб-програміст, веб-майстер тощо), хоча деякі веб-дизайнери можуть виконувати усе самостійно. Термін веб-дизайн зазвичай використовують для

опису проектування й реалізації клієнтської частини веб-сайту, включаючи верстк [6].

Для даної інформаційної сторінки був розроблений макет структури сторінок який зображено на рис. 3.7



Рисунок 3.7 – Макет структури сторінок інформаційної системи.

Головна сторінка сайту повинна містити коротку інформацію про сайт, але водночас не повинна перевантажувати зайвою інформацією. Сайт, а зокрема головна сторінка, повинні мати гарний дизайн, для того щоб зацікавити користувача. Проста та зрозуміла навігація для будь-якого користувача також не менш важлива деталь (рис. 3.8).

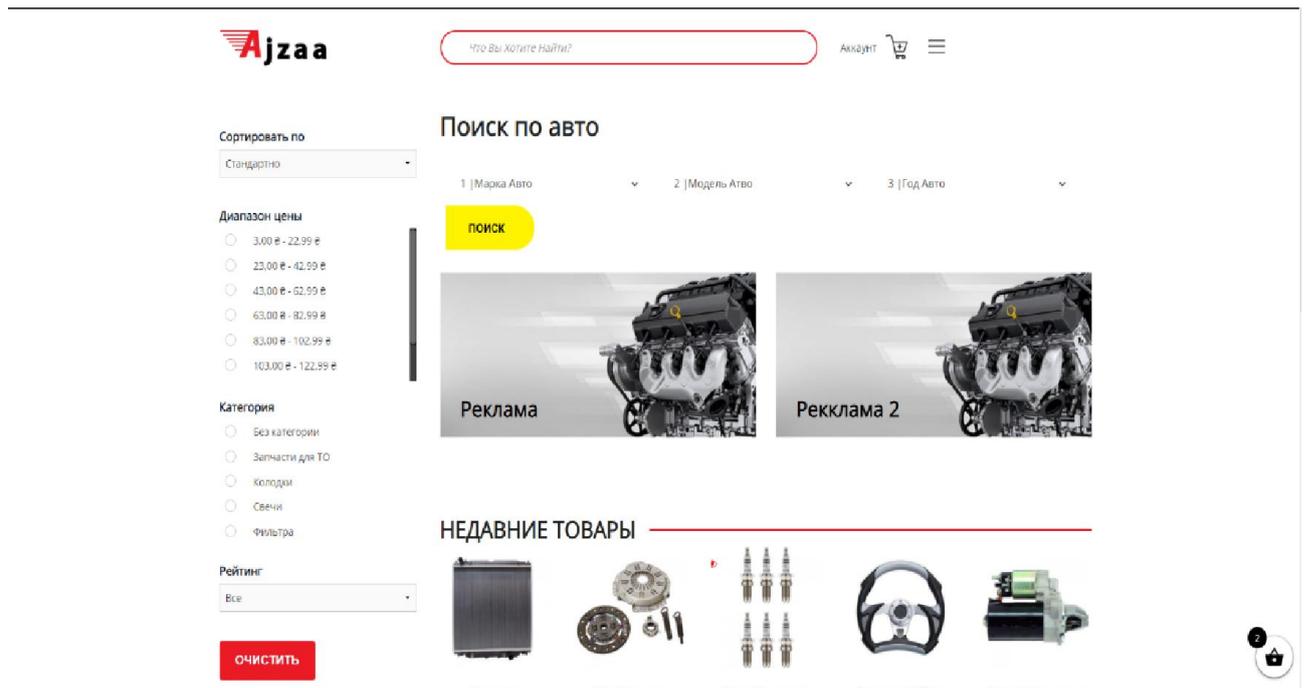


Рисунок 3.8 – Головна сторінка.

Головна сторінка містить наступні вкладки:

- новини;
- замовлення робіт;
- про компанію;
- контакти.

Також головна сторінка містить класифікацію послуг для автомобілів.

На інших сторінках розміщені послуги для автомобілі, які можна детально переглянути (рис. 3.9 – рис. 3-20), дізнатися цінові показники кожного з них, та замовити послуги, які вам найбільше підходять.



Рисунок 3.9 – Форма пошуку запасних частин по марці автомобіля

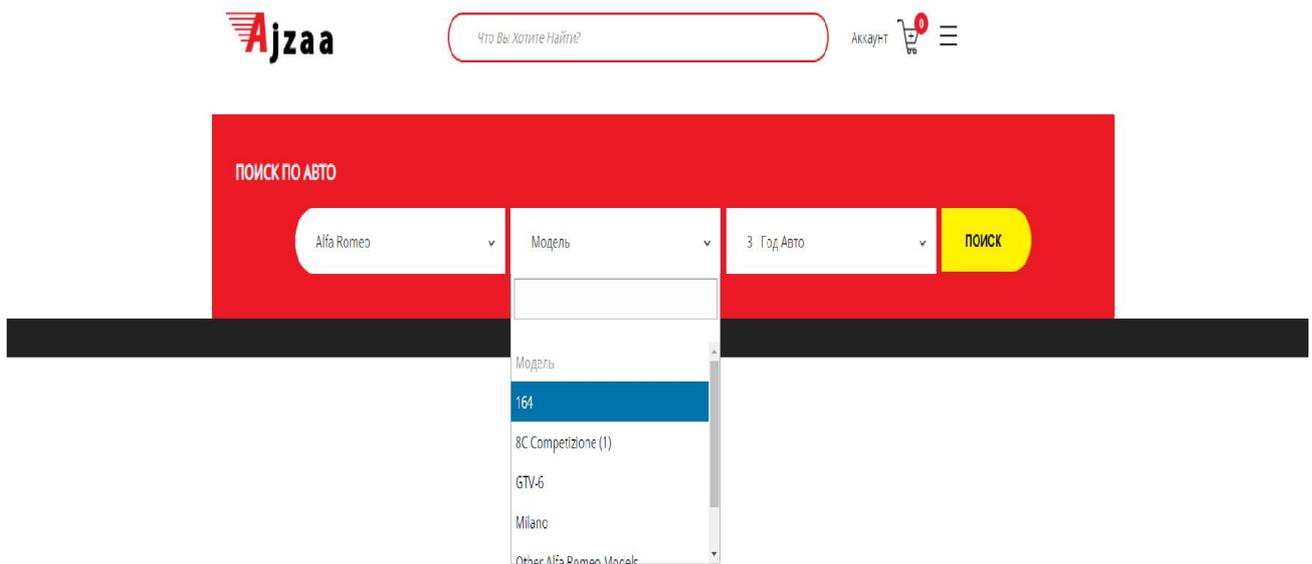


Рисунок 3.10 – Форма пошуку запасних частин по моделі автомобіля

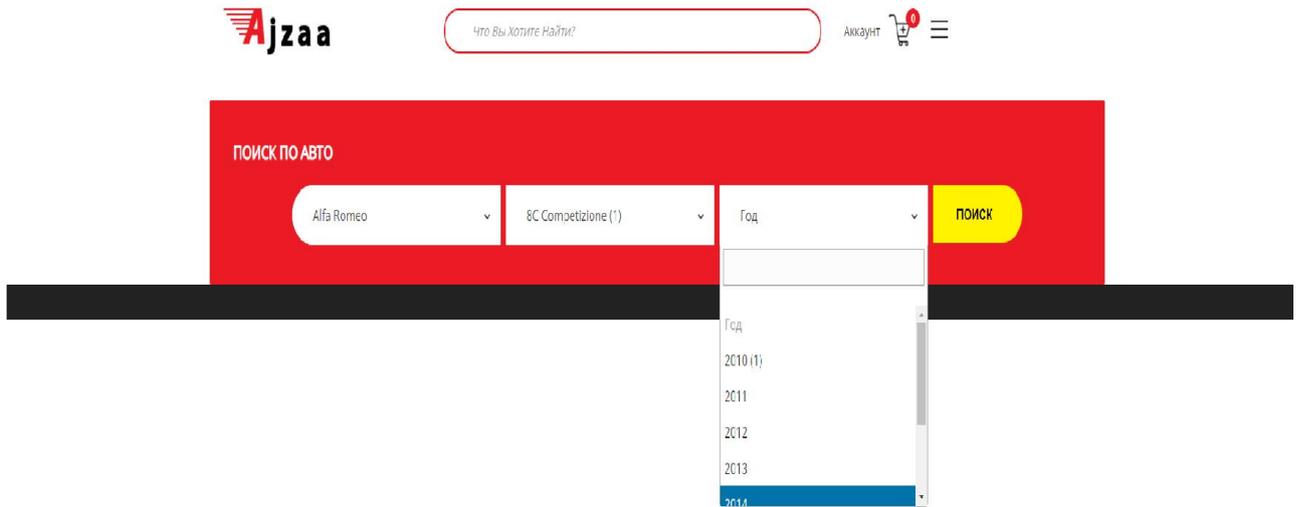


Рисунок 3.11 – Форма пошуку запасних частин по року автомобіля

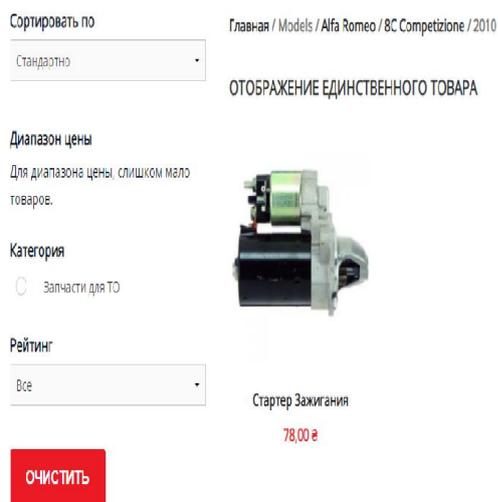


Рисунок 3.12 – Результат підбору

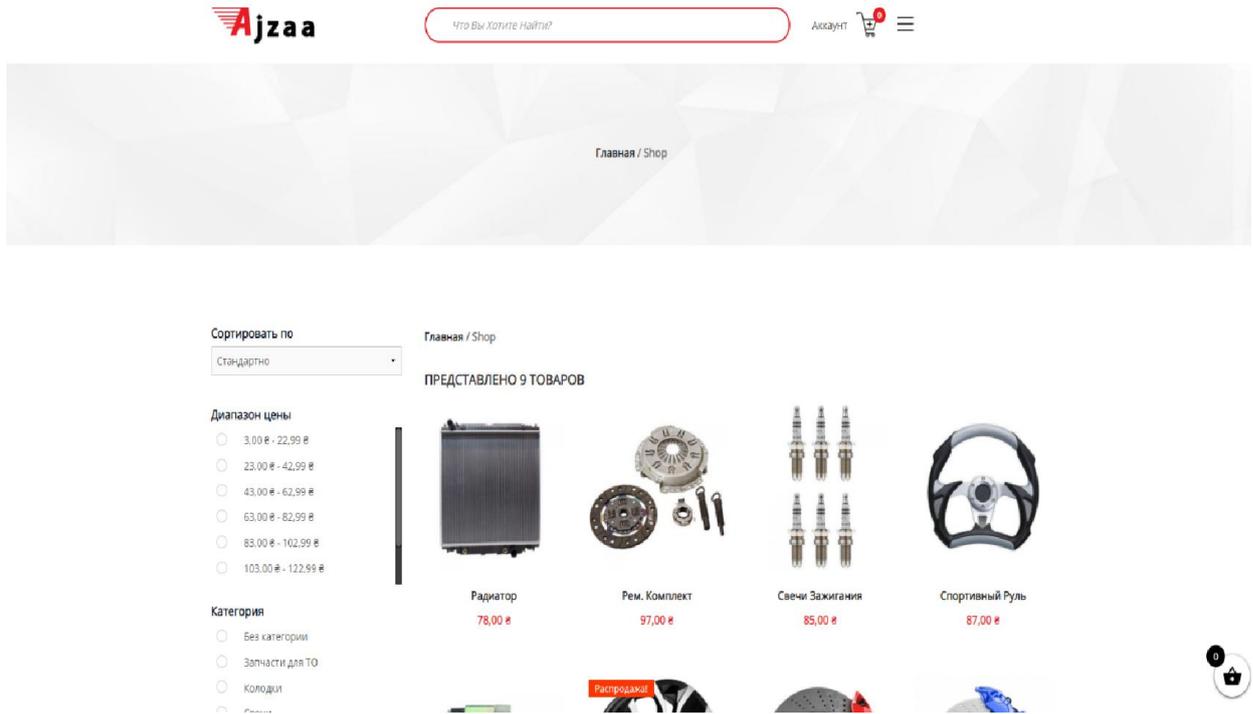


Рисунок 3.13 – Вивід всіх товарів

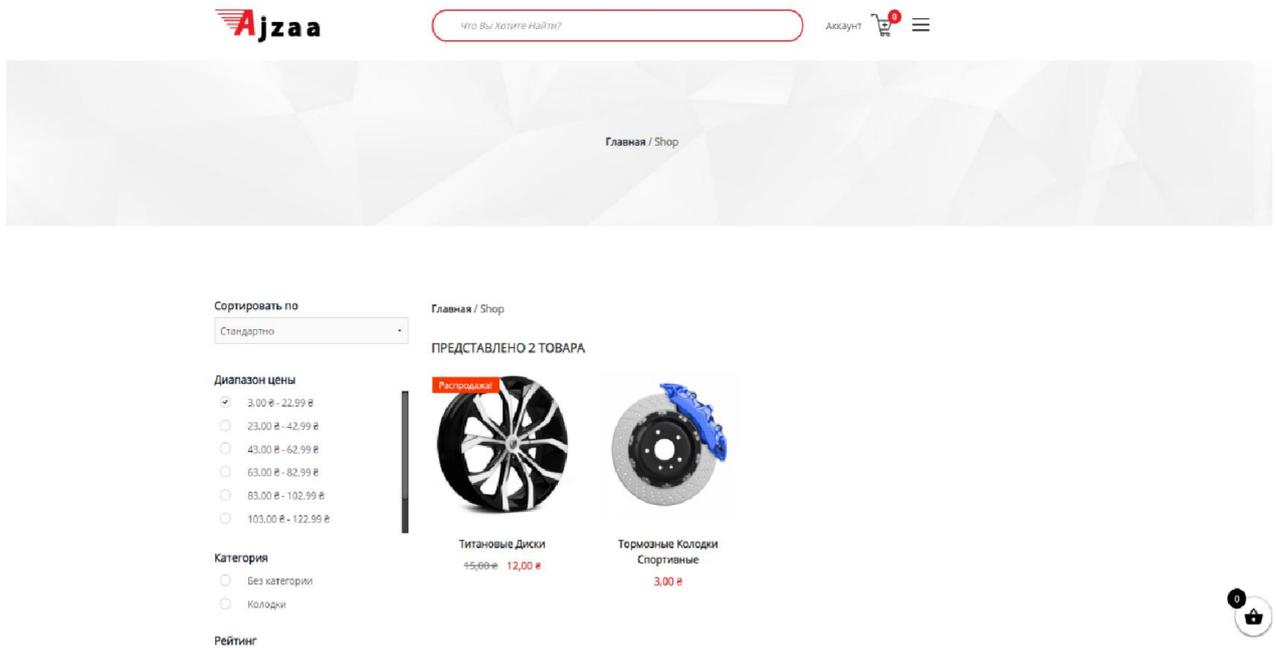


Рисунок 3.14 – Пошук товару

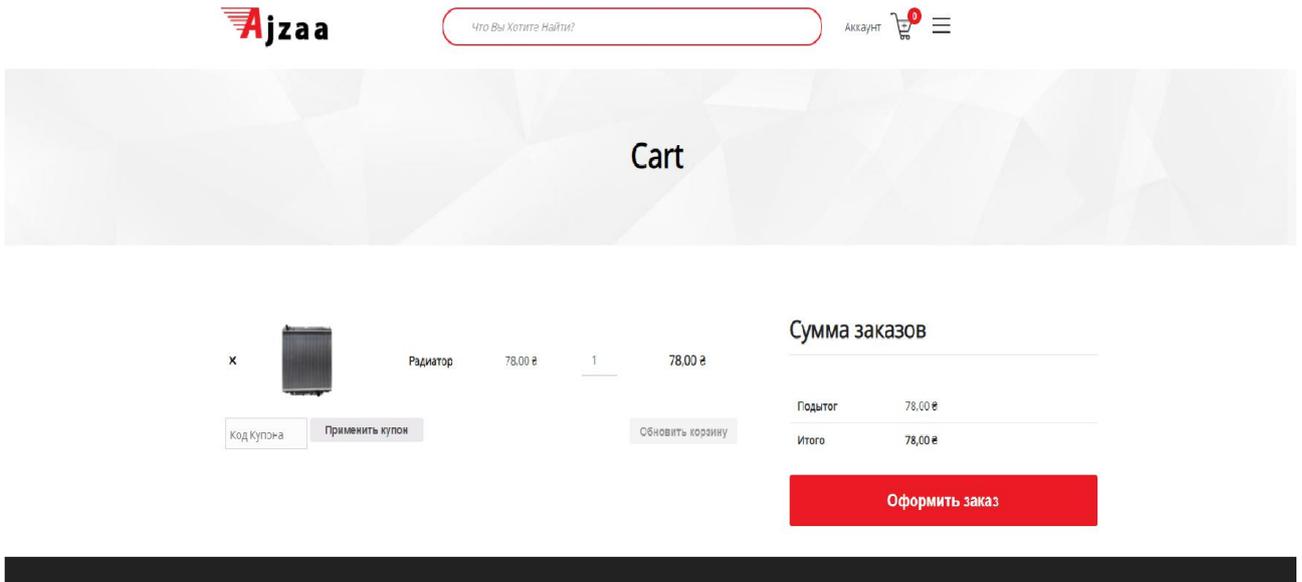


Рисунок 3.15 – Корзина выбранных товаров

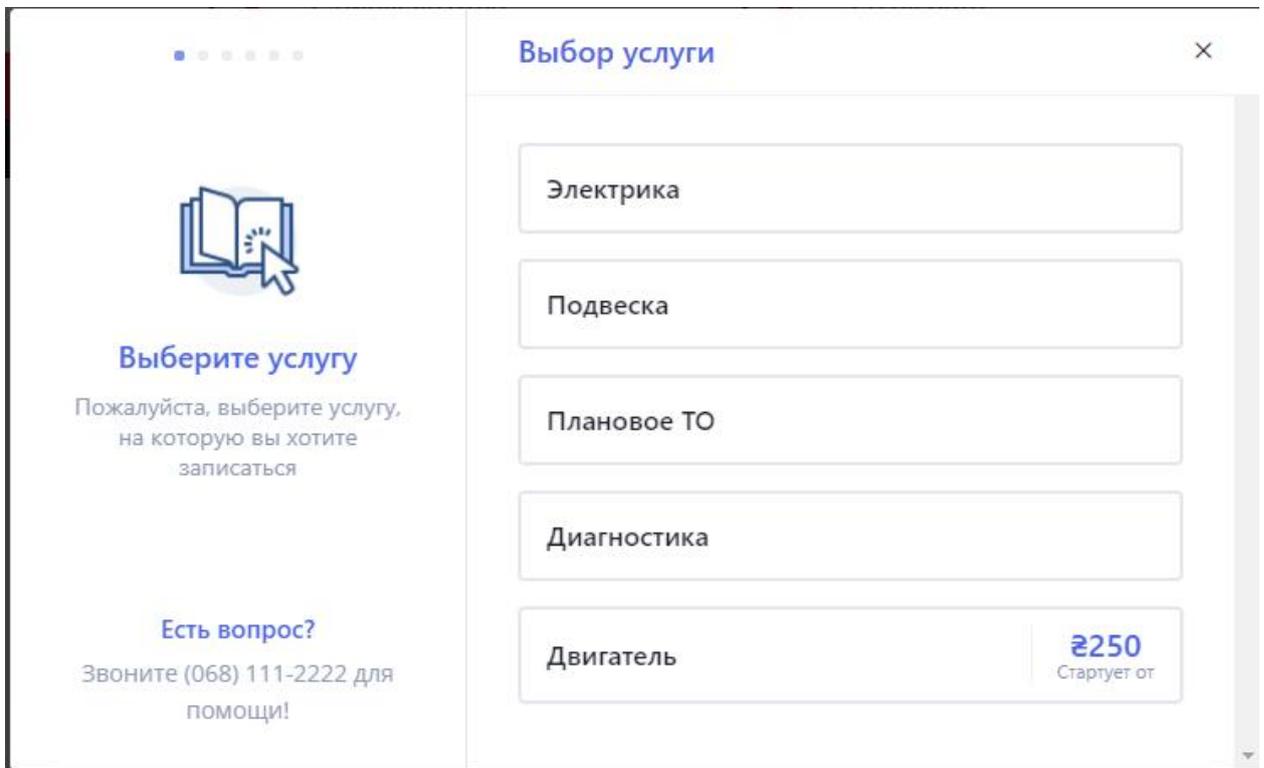


Рисунок 3.16 – Онлайн запись на обслуживание

Выбор даты и времени

Ноябрь 2021

ПН	ВТ	СР	ЧТ	ПТ	СБ	ВС
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Выберите **12:00** Ноябрь 17

8 9 10 11 12 13 14 15 16

← Назад Далее →

ОБЩЕЕ

УСЛУГА	ДАТА
Двигатель	Ноябрь 17
ВРЕМЯ	КЛИЕНТ
12:00	Миша Пупкин
ЦЕНА	
€333	

Выберите дату и время

Выберите дату на которую вы хотите записаться

Есть вопрос?
Звоните (068) 111-2222 для помощи!

Рисунок 3.17 – Онлайн запис на обслуговування на час

Информация о клиенте

Информация о клиенте Не вы? [Выйти](#)

Имя: Миша

Фамилия: Пупкин

Ваш телефон: (123) 123-1231

Email: Asdsad@Mail.Cpom

Добавить комментарий: Sadasdasd

← Назад Далее →

ОБЩЕЕ

УСЛУГА	ДАТА
Двигатель	Ноябрь 17
ВРЕМЯ	КЛИЕНТ
12:00	Миша Пупкин
ЦЕНА	
€333	

Введите информацию

Пожалуйста, предоставьте информацию что бы мы могли подтвердить вашу запись

Есть вопрос?
Звоните (068) 111-2222 для помощи!

Рисунок 3.18 – Інформація про клієнта

.....



Подтвердите вашу запись

Перепроверьте на корректность данных

Есть вопрос?
Звоните (068) 111-2222 для помощи!

Подтверждение ✕

Перепроверьте введенную вами информацию.

Запись

ДАТА: Ноя 24, 2021	ВРЕМЯ: 14:00
АДМИНИСТРАТОР: Admin Admin	УСЛУГА: Двигатель

Клиент

ИМЯ: Миша Пупкин	ТЕЛЕФОН: (123) 123-1231
EMAIL: asdsad@mail.cpom	

Цена

ПРИБЛИЗИТЕЛЬНАЯ ЦЕНА:
€333

← Назад
Отправить →

Рисунок 3.19 – Підтвердження про замовлення обслуговування клієнта

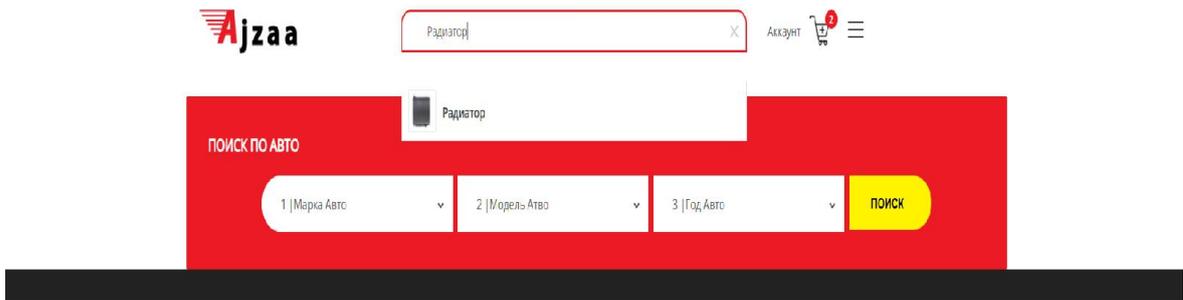


Рисунок 3.20 – Інтелектуальний пошук по сайту з виводом моментального результату

3.6 Вхід до панелі управління адміністратора

Для того, щоб потрапити до панелі інструментів адміністратора треба прописати в адресній строці <http://topcar.ls/auth/login>. Далі з'являється вікно, де потрібно ввести E-mail та пароль адміністратора (рис.3.21).

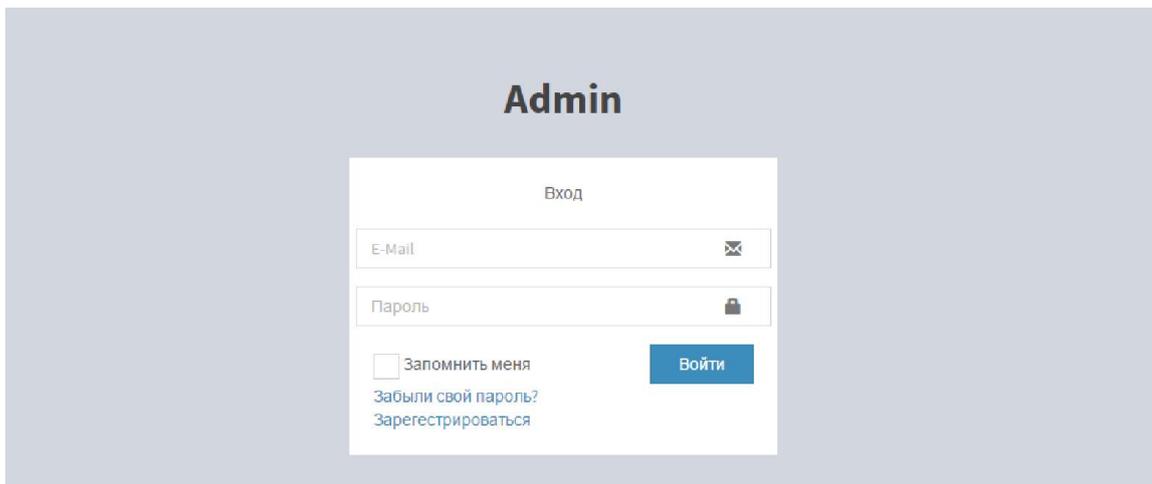


Рисунок 3.21 – Вхід до панелі управління адміністратора

Також за необхідністю можуть бути зареєстровані інші адміністратори. Для цього потрібно натиснути на кнопку «Зарегистрироваться» при авторизації адміністратора. Потім з'явиться вікно реєстрації, де потрібно заповнити всі необхідні поля для того, створити профіль. Вікно реєстрації має наступний вигляд (рис. 3.22).

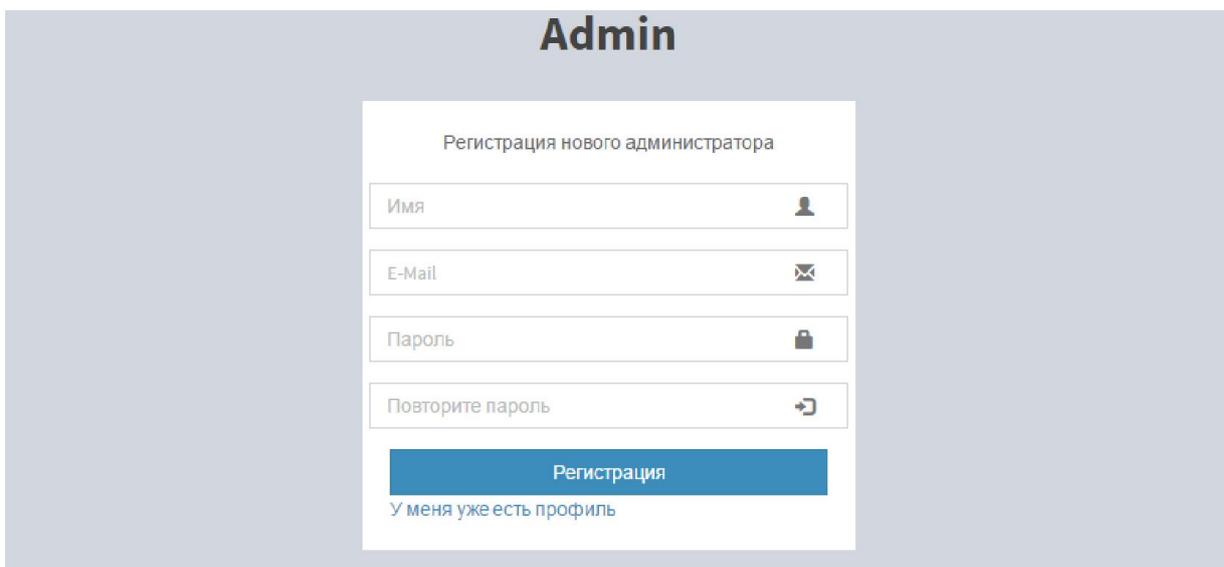


Рисунок 3.22– Реєстрація адміністратора

Панель управління адміністратора має наступні функції та можливості :

- додавання нових послуг для автомобілів;
- додавання новин;
- додавання нових адміністраторів;
- перегляд замовлень;
- відправка повідомлень про надання або відмову в наданні послуги;
- оповіщення користувачів про нові акції.

Панель управління адміністратора має бокове меню, за допомогою якого адміністратор зможе виконувати вище представлені функції (рис. 3.23).

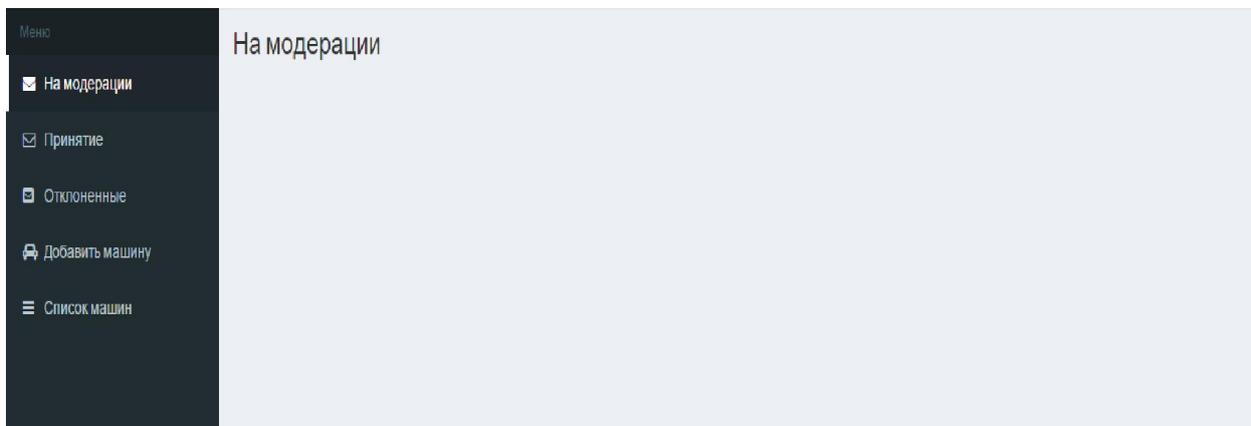


Рисунок 3.23– Панель управління адміністратора



Рисунок 3.24 – Останні записи на сторінці адміністратора

Показано с 1 по 4 из 4 Скачать CSV

ID	УСЛУГА	ДАТА	ВРЕМЯ	АДМИНИСТРАТОР	КЛИЕНТ	СТАТУС	СОЗДАНО	
4	Двигатель	Nov 24, 2021	14:00	Acmin Admin	Миша Пугин	Подтверждено	Nov 8, 2021	Редактировать
3	Подвеска	Oct 27, 2021	11:00	Acmin Admin	Миша Пугин	Подтверждено	Oct 24, 2021	Редактировать
2	Подвеска	Oct 28, 2021	14:00	Acmin Admin	Вася Пугин	Подтверждено	Oct 24, 2021	Редактировать
1	Подвеска	Oct 28, 2021	16:00	Acmin Admin	Вася Пугин	Подтверждено	Oct 24, 2021	Редактировать

Показано с 1 по 4 из 4 Page: 1

Рисунок 3.25 – Всі записи на сторінці адміністратора

Электрика	Подвеска	Плановое ТО	Диагностика	Двигатель	Добавить
Админ: [Avatar]	+				
Время: 60 мин					
Цена: 0	Цена: 0	Цена: 0	Цена: 0	Цена: €250	
Редактировать	Редактировать	Редактировать	Редактировать	Редактировать	

Рисунок 3.26 – Послуги на сторінці адміністратора

Информация

Название	Описание
Видимость	Статус
Видно всем	Активно

Цена и Время

Среднее время работы (минуты)	Цена
60	

Отображаемая цена

Эта цена будет отображаться, а не взиматься.

Мин. Цена	Макс. Цена

Рисунок 3.27 – Додавання та редагування послуг на сторінці адміністратора

Введите для поиска услуг, администратора, клиента...

БЛИЖАЙШИЕ ЗАПИСИ
ЗАПИСИ

ДАННЫЕ

УСЛУГИ
АДМИНИСТРАТОРЫ
КЛИЕНТЫ

Показано с 1 по 2 из 2 [Скачать CSV](#)

ID	ИМЯ	ТЕЛЕФОН	EMAIL	ЗАПИСЕЙ	БЛИЖАЙШАЯ ЗАПИСЬ	ВРЕМЕНИ ДО СЛЕД.	ЗАРЕГИСТРИРОВАН	
ID	По имени	Телефон...	По Email				По date	
2	Миша Пупкин	1231231231	asdsad@mail.com	2	November 24, 14:00	16 Дней	Oct 24, 2021	Редактировать
1	Вася Пупкин	1232132131	adasadsdsasd@mail.ru	2	n/a	Past	Oct 24, 2021	Редактировать
ID	ИМЯ	ТЕЛЕФОН	EMAIL	ЗАПИСЕЙ	БЛИЖАЙШАЯ ЗАПИСЬ	ВРЕМЕНИ ДО СЛЕД.	ЗАРЕГИСТРИРОВАН	

Показано с 1 по 2 из 2 Страница: 1

Рисунок 3.28 – Зарегестровані клієнти на сторінці адміністратора

РОЗДІЛ 4 ТЕСТУВАННЯ

4.1 Тестування ІС

Якість програмного продукту характеризується набором властивостей, що визначають, наскільки продукт «хороший» 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 з найбільш усталених способів забезпечення якості розробки програмного забезпечення і входить в набір ефективних засобів сучасної системи забезпечення якості програмного продукту [6].

З технічної точки зору тестування полягає у виконанні програми на деякій множині вихідних даних м збірці одержуваних результатів із заздалегідь відомими (еталонними) з метою встановити відповідність різних властивостей і характеристик програми замовленим властивостями.

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:

1. що треба тестувати (об'єкт тестування: система, додаток, обладнання);
2. що будете тестувати (список функцій і компонент тестується системи);
3. як будете тестувати (стратегія тестування - види тестування і їх застосування по відношенню до тестованого об'єкту).

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

Таблиця 4.1 – Тестування верстки

	Mozilla Firefox	Google Chrome	Internet Explorer
Коректне відображення картинок	Успішно	Успішно	Успішно
Клікабельність посилань	Успішно	Успішно	Успішно
Відображення колірної гами 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

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

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

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	З'явилося вікно 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	З'явилося вікно 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
Тест-кейс №3. Тестування при вірному введенні логіна та пароля	
Дії	Очікуваний результат
1. Відкриваємо інформаційну систему від імені адміністратора	З'явилося вікно 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

ВИСНОВКИ

В даній дипломній роботі була спроектована та програмно реалізована інформаційна система з надання послуг автосервісу м. Полтава на базі підприємства «AJZAA». 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 бронювання послуг для автомобілів;
- швидке та якісне обслуговування;
- простота доступу та замовлення авто, завдяки відсутності реєстрації;
- 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 обрані послуги на автомобіль.

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

Таким чином, впровадження цього проекту значною мірою поліпшує збереження цілісності даних, перегляд та пошук необхідної інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Анісімов А.В. Інформаційні системи та бази даних: Навчальний посібник для студентів факультету комп'ютерних наук та кібернетики. / Анісімов А.В., Кулябко П.П. – Київ. – 2017. – 110 с.
2. Антоненко В. М. Сучасні інформаційні системи і технології: управління знаннями : навч. посібник / В. М. Антоненко, С. Д. Мамченко, Ю. В. Рогущина. – Ірпінь : Нац. університет ДПС України, 2016. – 212 с.
3. Воронін А. М. Інформаційні системи прийняття рішень: навчальний посібник. / Воронін А. М., Зіатдінов Ю. К., Климова А. С. – К. : НАУ-друк, 2009. – 136с.
4. Галузинський Г. П. Інформаційні системи у бізнесі. Практикум для індивідуальної роботи: навч.- метод. посіб. для самост. вивч. Дисципліни. / Галузинський Г. П., Денісова О. О., Писаревська Т. А. – К. : КНЕУ, 2008. – 524с.
5. Годун В.М. Інформаційні системи і технології в статистиці: навч. посіб. / В.М. Годун, Н.С. Орленко, М. А. Сендзюк; за ред. В.Ф. Ситника. – К.: КНЕУ, 2003. – 267 с
Комп'ютерні та Інформаційні технології і системи [Електронний ресурс] – режим доступу: <http://fitts.pntu.edu.ua/ua/kafedry/inf-tekh-i-sy-m>
6. Полтавський національний технічний університет імені Юрія Кондратюка [Електронний ресурс] – режим доступу: <http://pntu.edu.ua/>
7. Сайт харківського національного університету радіо-електроніки [Електронний ресурс] – режим доступу: <http://nure.ua/uk/>
8. Сайт київського політехнічного інституту імені Ігоря Сікорського [Електронний ресурс] – режим доступу: <http://kpi.ua/ru>
9. XHTML [Електронний ресурс] – режим доступу: <http://htmlbook.ru/xhtml>
10. ANGULAR [Електронний ресурс] – режим доступу: <https://metanit.com/web/angular/>

11. АРАСНЕ [Електронний ресурс] – режим доступу: <http://www.apache.ru/>
12. MYSQL [Електронний ресурс] – режим доступу: <https://www.mysql.com/>
13. Діаграма прецедентів [Електронний ресурс] – режим доступу: [http://www.planerka.info/item/Diagramma-precedentov-\(variantov-ispolzovaniya\)-UML](http://www.planerka.info/item/Diagramma-precedentov-(variantov-ispolzovaniya)-UML)
14. Діаграма діяльності [Електронний ресурс] – режим доступу: http://it-gost.ru/articles/view_articles/96
15. Контекстна діаграма [Електронний ресурс] – режим доступу: <http://techtrend.com.ua/index.php>
16. Діаграма декомпозиції [Електронний ресурс] – режим доступу: http://studopedia.com.ua/1_162873_diagrami-dekompozitsii.html
17. Розробка інформаційної системи [Електронний ресурс] – режим доступу: http://bukvar.su/informatika_programmirovanie
18. Аналіз баз даних [Електронний ресурс] – режим доступу: <https://www.osp.ru/os/2002/03/181272/>
19. Порівняльний аналіз СУБД [Електронний ресурс] – режим доступу: <https://sibac.info/studconf/tech/xxxvi/43413>
20. Порівняльний аналіз серверів [Електронний ресурс] – режим доступу: <http://www.it-ic.ru/sravnenie-veb-serverov-apache-iis>
21. Огляд web-серверів [Електронний ресурс] – режим доступу: <http://info-comp.ru/sisadminst/221-popular-web-servers.html>

ДОДАТОК А

КОД ПРОГРАМИ

```

<?php
$pix_layout = pixtheme_get_layout( get_post_type(), get_the_ID() );
$pix_width_class = pixtheme_get_layout_width( get_post_type() );

?>

<?php get_header();?>
    <section class="blog" >
        <div class="container <?php echo esc_attr($pix_width_class)?>">
            <div class="row">

                <?php pixtheme_show_sidebar( 'left', $pix_layout['layout'],
$pix_layout['sidebar'] ); ?>

                <div class="<?php if ( $pix_layout['layout'] != 1 ) : ?>col-xl-9
col-lg-8 <?php endif; ?> col-12 <?php echo esc_attr($pix_layout['class'])?>">

                    <div class="rtd"> <?php if ( have_posts() ) while (
have_posts() ) : the_post(); ?>
                        <?php $pixtheme_page_com_id = $post; ?>
                        <?php the_content(); ?>
                        <?php
                            $args = array(
                                'before' => '<p class="pix-link-
pages"><span>'.esc_html__( 'Pages:', 'solutech' ).'</span>',
                                'after' => '</p>',
                            );
                            wp_link_pages($args);
                        ?>
                        <?php
                            if( 'open' == $pixtheme_page_com_id->comment_status) {
                                comments_template();
                            }
                        ?>
                        <?php endwhile; ?>
                    </div>
                </div>

                <?php pixtheme_show_sidebar( 'right', $pix_layout['layout'],
$pix_layout['sidebar'] ); ?>

            </div>
        </div>
    </section>
<?php get_footer(); ?>

```

Index.php

```

<?php
$pix_layout = pixtheme_get_layout('blog');
$pix_width_class = pixtheme_get_layout_width( get_post_type() );

$blog_class = 'pix-blog-' . pixtheme_get_setting('pix-blog-style', 'classic');

?>
<?php get_header();?>

```

```

<section class="blog">
    <div class="container <?php echo esc_attr($pix_width_class)?>">
        <div class="row sidebar-type-<?php echo esc_attr($pix_layout['layout']);
?>">
            <?php pixtheme_show_sidebar( 'left', $pix_layout['layout'],
$pix_layout['sidebar'] ); ?>

            <div class="<?php if ( $pix_layout['layout'] != 1 ) : ?>col-xl-9
col-lg-8<?php endif; ?> col-12 <?php echo esc_attr($pix_layout['class'])?>">
                <section class="<?php echo esc_attr($blog_class) ?>">
                    <?php
                        $wp_query = new WP_Query();
                        $pp = get_option('posts_per_page');
                        $wp_query->query('posts_per_page='.$pp.'&paged='.$paged);
                        get_template_part( 'loop', 'index' );
                    ?>
                </section>
                <?php
                    the_posts_pagination( array(
                        'prev_text' => wpkses_post__( '<i class="fas
fa-chevron-left"></i>', 'solutech' ),
                        'next_text' => wpkses_post__( '<i class="fas
fa-chevron-right"></i>', 'solutech' ),
                        'screen_reader_text' => esc_html__( '&nbsp;',
'solutech' ),
                    ) );
                ?>
            </div>
            <!-- end col -->

            <?php pixtheme_show_sidebar( 'right', $pix_layout['layout'],
$pix_layout['sidebar'] ); ?>
        </div>
    <!-- end row -->
</div>
</section>
<?php get_footer(); ?>

```

Service-cat.php

```

<?php
/** The taxonomy for services category. */

$pix_layout = pixtheme_get_layout('service-cat');
$pix_width_class = pixtheme_get_layout_width( 'service-cat' );

$pix_term = get_term_by( 'slug', get_query_var( 'term' ), get_query_var(
'taxonomy' ) );
$services_page = ''; //pixtheme_get_setting('pix-page-service-cat', '');
$all_services = '';
if ( '' != $services_page ) {
    $all_services = get_the_permalink($services_page);
}

$style = 'bottom-info';

$pix_col = pixtheme_get_setting('pix-col-service-cat', '3');
$pix_gap = '50';

$image_size = $image_class = pixtheme_get_setting('pix-images-service-cat', '');
if($pix_col == 3){
    $image_size .= '-col-3';
} elseif($pix_col > 3) {

```



```

        if ( ! empty($cats) ) {
            foreach ( $cats as $cat ) {
                $pixtheme_cat_slugs .= $cat->slug . " ";
            }
        }
        $thumbnail = get_the_post_thumbnail(get_the_ID(),
$image_size, array('class' => $image_class.' pix-img-greyscale'));

        // category list linked
        $pix_linked_list_cats = pixtheme_get_post_terms( array(
'taxonomy' => 'pix-service-cat', 'items_wrap' => '%s' ) );
        ?>

        <div class="pix-isotope-item pix-box">

            <?php
                if( $style == 'hover-info' ){
                    $out_content = `
                        <div class="pix-hover-container">
                            <h3><a href="" ` .
esc_url(get_the_permalink()) . "`>` . wp_kses_post(get_the_title()) . `</a></h3>
                            <p>` . wp_kses_post($pix_linked_list_cats) .
                            `</p>
                        </div>
                    ` . wp_kses_post($thumbnail);

                } elseif($style == 'bottom-desc') {
                    $out_content = `
                        <div class="pix-box-img">
                            <a href="" ` . esc_url(get_the_permalink()) .
                            `>
                                <div class="pix-img-wrapper">` .
wp_kses_post($thumbnail) . `</div>
                                </a>
                            </div>
                            <div class="pix-box-text">
                                <div class="pix-box-name">
                                    <a href="" ` .
esc_url(get_the_permalink()) . "`>` . wp_kses_post(get_the_title()) . `</a>
                                </div>
                                ` .pixtheme_limit_words(get_the_excerpt(),
20, `p`).'
                            </div>
                            <span class=""><a
href=""`.esc_url(get_permalink(get_the_ID())).`></a></span>`;
                    } else {

                        $out_content = `
                            <div class="pix-box-img">
                                <a href="" ` . esc_url(get_the_permalink()) .
                                `>
                                    ` . wp_kses_post($thumbnail) . `
                                </a>
                            </div>
                            <div class="pix-box-name">
                                <a href="" ` . esc_url(get_the_permalink()) .
                                `>` . wp_kses_post(get_the_title()) . `</a>
                            </div>`;
                    }
                    echo wp_kses_post($out_content);
                }
            ?>

        </div>

```

```

        <?php
            endwhile;
        ?>
    </div>
</div>

<?php
endif;
?>

<?php
$t_slug = $pix_term->slug;
$cat_meta = get_option("pix-service-cat_{$t_slug}");
$pix_section_id = in_array($cat_meta['pix_cat_section'],
array('global', '')) ? '' : $cat_meta['pix_cat_section'];
if(function_exists('icl_object_id')) {
    $pix_section_id = icl_object_id ($pix_section_id,'pix-
section',true);
}
if ( $pix_section_id ) {
    pixtheme_get_section_content($pix_section_id);
}
?>

</div>

<?php pixtheme_show_sidebar( 'right', $pix_layout['layout'],
$pix_layout['sidebar'] ); ?>

</div>
</div>
</section>

<?php get_footer() ?>

```

Pix-service

```

<?php /* Template Name: Single Service */

$pix_layout = pixtheme_get_layout(get_post_type(), get_the_ID());
$pix_width_class = pixtheme_get_layout_width( get_post_type() );

get_header();
?>

<section class="blog service">
    <div class="container <?php echo esc_attr($pix_width_class)?>">
        <div class="row">

            <?php pixtheme_show_sidebar( 'left', $pix_layout['layout'],
$pix_layout['sidebar'] ); ?>

            <div class="<?php if ( $pix_layout['layout'] != 1 ) : ?>col-xl-9
col-lg-8<?php endif; ?> col-12 <?php echo esc_attr($pix_layout['class']); ?>">
                <div class="rtd">
                    <?php if ( have_posts() ) while ( have_posts() ) : the_post();
?>
                        <?php the_content(); ?>
                    <?php endwhile; ?>
                </div>
            </div>
        </div>
    </div>

```

```

    <?php pixtheme_show_sidebar( 'right', $pix_layout['layout'],
    $pix_layout['sidebar'] ); ?>

    </div>
  </div>

```

```
</section>
```

```
<?php get_footer();?>
```

Script.js

```

document.getElementById('forCrip')
    .addEventListener('change', function() {

    var fr=new FileReader();
    fr.onload=function(){
        document.getElementById('texto')
            .textContent=fr.result;
    }

    fr.readAsText(this.files[0]);
})

document.getElementById('forDecrip')
    .addEventListener('change', function() {

    var fr=new FileReader();
    fr.onload=function(){
        document.getElementById('dec_texto')
            .textContent=fr.result;
    }

    fr.readAsText(this.files[0]);
})

var Aes = {}; // Aes namespace

Aes.cipher = function(input, w) {
    var Nb = 4; // block size (in words): no of columns in state
    (fixed at 4 for AES)
    var Nr = w.length/Nb - 1; // no of rounds: 10/12/14 for 128/192/256-bit keys

    var state = [[],[],[],[ ]]; // initialise 4xNb byte-array 'state' with input
    [§3.4]
    for (var i=0; i<4*Nb; i++) state[i%4][Math.floor(i/4)] = input[i];

    state = Aes.addRoundKey(state, w, 0, Nb);

    for (var round=1; round<Nr; round++) {
        state = Aes.subBytes(state, Nb);
        state = Aes.shiftRows(state, Nb);
        state = Aes.mixColumns(state, Nb);
        state = Aes.addRoundKey(state, w, round, Nb);
    }

    state = Aes.subBytes(state, Nb);
    state = Aes.shiftRows(state, Nb);
    state = Aes.addRoundKey(state, w, Nr, Nb);

```

```

    var output = new Array(4*Nb); // convert state to 1-d array before returning
    [S3.4]
    for (var i=0; i<4*Nb; i++) output[i] = state[i%4][Math.floor(i/4)];
    return output;
}

/**
 * Perform Key Expansion to generate a Key Schedule
 *
 * @param {Number[]} key Key as 16/24/32-byte array
 * @returns {Number[][]} Expanded key schedule as 2D byte-array (Nr+1 x Nb
bytes)
 */
Aes.keyExpansion = function(key) { // generate Key Schedule (byte-array Nr+1 x
Nb) from Key [S5.2]
    var Nb = 4; // block size (in words): no of columns in state (fixed
at 4 for AES)
    var Nk = key.length/4 // key length (in words): 4/6/8 for 128/192/256-bit
keys
    var Nr = Nk + 6; // no of rounds: 10/12/14 for 128/192/256-bit keys

    var w = new Array(Nb*(Nr+1));
    var temp = new Array(4);

    for (var i=0; i<Nk; i++) {
        var r = [key[4*i], key[4*i+1], key[4*i+2], key[4*i+3]];
        w[i] = r;
    }

    for (var i=Nk; i<(Nb*(Nr+1)); i++) {
        w[i] = new Array(4);
        for (var t=0; t<4; t++) temp[t] = w[i-1][t];
        if (i % Nk == 0) {
            temp = Aes.subWord(Aes.rotWord(temp));
            for (var t=0; t<4; t++) temp[t] ^= Aes.rCon[i/Nk][t];
        } else if (Nk > 6 && i%Nk == 4) {
            temp = Aes.subWord(temp);
        }
        for (var t=0; t<4; t++) w[i][t] = w[i-Nk][t] ^ temp[t];
    }

    return w;
}

Aes.subBytes = function(s, Nb) { // apply SBox to state S [S5.1.1]
    for (var r=0; r<4; r++) {
        for (var c=0; c<Nb; c++) s[r][c] = Aes.sBox[s[r][c]];
    }
    return s;
}

Aes.shiftRows = function(s, Nb) {
    var t = new Array(4);
    for (var r=1; r<4; r++) {
        for (var c=0; c<4; c++) t[c] = s[r][(c+r)%Nb];
        for (var c=0; c<4; c++) s[r][c] = t[c];
    }
    return s;
}

Aes.mixColumns = function(s, Nb) { // S [S5.1.3]
    for (var c=0; c<4; c++) {

```

```

var a = new Array(4); // 'a' is a copy of the current column from 's'
var b = new Array(4); // 'b' is a·{02} in GF(2^8)
for (var i=0; i<4; i++) {
    a[i] = s[i][c];
    b[i] = s[i][c]&0x80 ? s[i][c]<<1 ^ 0x011b : s[i][c]<<1;
}
// a[n] ^ b[n] is a·{03} in GF(2^8)
s[0][c] = b[0] ^ a[1] ^ b[1] ^ a[2] ^ a[3]; // 2*a0 + 3*a1 + a2 + a3
s[1][c] = a[0] ^ b[1] ^ a[2] ^ b[2] ^ a[3]; // a0 * 2*a1 + 3*a2 + a3
s[2][c] = a[0] ^ a[1] ^ b[2] ^ a[3] ^ b[3]; // a0 + a1 + 2*a2 + 3*a3
s[3][c] = a[0] ^ b[0] ^ a[1] ^ a[2] ^ b[3]; // 3*a0 + a1 + a2 + 2*a3
}
return s;
}

Aes.addRoundKey = function(state, w, rnd, Nb) { // xor Round Key into state S
    [5.1.4]
    for (var r=0; r<4; r++) {
        for (var c=0; c<Nb; c++) state[r][c] ^= w[rnd*4+c][r];
    }
    return state;
}

Aes.subWord = function(w) { // apply SBox to 4-byte word w
    for (var i=0; i<4; i++) w[i] = Aes.sBox[w[i]];
    return w;
}

Aes.rotWord = function(w) { // rotate 4-byte word w left by one byte
    var tmp = w[0];
    for (var i=0; i<3; i++) w[i] = w[i+1];
    w[3] = tmp;
    return w;
}

Aes.sBox =
[0x63,0x7c,0x77,0x7b,0xf2,0x6b,0x6f,0xc5,0x30,0x01,0x67,0x2b,0xfe,0xd7,0xab,0x76
,
0xca,0x82,0xc9,0x7d,0xfa,0x59,0x47,0xf0,0xad,0xd4,0xa2,0xaf,0x9c,0xa4,0x72,0xc0,
0xb7,0xfd,0x93,0x26,0x36,0x3f,0xf7,0xcc,0x34,0xa5,0xe5,0xf1,0x71,0xd8,0x31,0x15,
0x04,0xc7,0x23,0xc3,0x18,0x96,0x05,0x9a,0x07,0x12,0x80,0xe2,0xeb,0x27,0xb2,0x75,
0x09,0x83,0x2c,0x1a,0x1b,0x6e,0x5a,0xa0,0x52,0x3b,0xd6,0xb3,0x29,0xe3,0x2f,0x84,
0x53,0xd1,0x00,0xed,0x20,0xfc,0xb1,0x5b,0x6a,0xcb,0xbe,0x39,0x4a,0x4c,0x58,0xcf,
0xd0,0xef,0xaa,0xfb,0x43,0x4d,0x33,0x85,0x45,0xf9,0x02,0x7f,0x50,0x3c,0x9f,0xa8,
0x51,0xa3,0x40,0x8f,0x92,0x9d,0x38,0xf5,0xbc,0xb6,0xda,0x21,0x10,0xff,0xf3,0xd2,
0xcd,0x0c,0x13,0xec,0x5f,0x97,0x44,0x17,0xc4,0xa7,0x7e,0x3d,0x64,0x5d,0x19,0x73,
0x60,0x81,0x4f,0xdc,0x22,0x2a,0x90,0x88,0x46,0xee,0xb8,0x14,0xde,0x5e,0x0b,0xdb,
0xe0,0x32,0x3a,0x0a,0x49,0x06,0x24,0x5c,0xc2,0xd3,0xac,0x62,0x91,0x95,0xe4,0x79,
0xe7,0xc8,0x37,0x6d,0x8d,0xd5,0x4e,0xa9,0x6c,0x56,0xf4,0xea,0x65,0x7a,0xae,0x08,
0xba,0x78,0x25,0x2e,0x1c,0xa6,0xb4,0xc6,0xe8,0xdd,0x74,0x1f,0x4b,0xbd,0x8b,0x8a,

```

```

0x70,0x3e,0xb5,0x66,0x48,0x03,0xf6,0x0e,0x61,0x35,0x57,0xb9,0x86,0xc1,0x1d,0x9e,
0xe1,0xf8,0x98,0x11,0x69,0xd9,0x8e,0x94,0x9b,0x1e,0x87,0xe9,0xce,0x55,0x28,0xdf,
0x8c,0xa1,0x89,0x0d,0xbf,0xe6,0x42,0x68,0x41,0x99,0x2d,0x0f,0xb0,0x54,0xbb,0x16]
;

```

```

Aes.rCon = [ [0x00, 0x00, 0x00, 0x00],
              [0x01, 0x00, 0x00, 0x00],
              [0x02, 0x00, 0x00, 0x00],
              [0x04, 0x00, 0x00, 0x00],
              [0x08, 0x00, 0x00, 0x00],
              [0x10, 0x00, 0x00, 0x00],
              [0x20, 0x00, 0x00, 0x00],
              [0x40, 0x00, 0x00, 0x00],
              [0x80, 0x00, 0x00, 0x00],
              [0x1b, 0x00, 0x00, 0x00],
              [0x36, 0x00, 0x00, 0x00] ];

```

```
Aes.Ctr = {};
```

```

Aes.Ctr.encrypt = function(plaintext, password, nBits) {
    var blockSize = 16;
    if (!(nBits==128 || nBits==192 || nBits==256)) return '';
    plaintext = Utf8.encode(plaintext);
    password = Utf8.encode(password);

    var nBytes = nBits/8;
    var pwBytes = new Array(nBytes);
    for (var i=0; i<nBytes; i++) {
        pwBytes[i] = isNaN(password.charCodeAt(i)) ? 0 : password.charCodeAt(i);
    }
    var key = Aes.cipher(pwBytes, Aes.keyExpansion(pwBytes));
    key = key.concat(key.slice(0, nBytes-16));
    var counterBlock = new Array(blockSize);

    var nonce = (new Date()).getTime();
    var nonceMs = nonce%1000;
    var nonceSec = Math.floor(nonce/1000);
    var nonceRnd = Math.floor(Math.random()*0xffff);

    for (var i=0; i<2; i++) counterBlock[i] = (nonceMs >>> i*8) & 0xff;
    for (var i=0; i<2; i++) counterBlock[i+2] = (nonceRnd >>> i*8) & 0xff;
    for (var i=0; i<4; i++) counterBlock[i+4] = (nonceSec >>> i*8) & 0xff;

    var ctrTxt = '';
    for (var i=0; i<8; i++) ctrTxt += String.fromCharCode(counterBlock[i]);

    var keySchedule = Aes.keyExpansion(key);

    var blockCount = Math.ceil(plaintext.length/blockSize);
    var ciphertxt = new Array(blockCount); // ciphertext as array of strings

    for (var b=0; b<blockCount; b++) {

        for (var c=0; c<4; c++) counterBlock[15-c] = (b >>> c*8) & 0xff;
        for (var c=0; c<4; c++) counterBlock[15-c-4] = (b/0x100000000 >>> c*8)

        var cipherCntr = Aes.cipher(counterBlock, keySchedule);

```

```

    var blockLength = b < blockCount - 1 ? blockSize : (plaintext.length -
1) % blockSize + 1;
    var cipherChar = new Array(blockLength);

    for (var i=0; i < blockLength; i++) { // -- xor plaintext with ciphered
counter char-by-char --
        cipherChar[i] = cipherCtr[i] ^ plaintext.charCodeAt(b * blockSize + i);
        cipherChar[i] = String.fromCharCode(cipherChar[i]);
    }
    ciphertxt[b] = cipherChar.join('');
}

var ciphertext = ctrTxt + ciphertxt.join('');
ciphertext = Base64.encode(ciphertext); // encode in base64

return ciphertext;
}

Aes.Ctr.decrypt = function(ciphertext, password, nBits) {
    var blockSize = 16;
    if (!(nBits == 128 || nBits == 192 || nBits == 256)) return '';
    ciphertext = Base64.decode(ciphertext);
    password = Utf8.encode(password);

    var nBytes = nBits / 8;
    var pwBytes = new Array(nBytes);
    for (var i=0; i < nBytes; i++) {
        pwBytes[i] = isNaN(password.charCodeAt(i)) ? 0 : password.charCodeAt(i);
    }
    var key = Aes.cipher(pwBytes, Aes.keyExpansion(pwBytes));
    key = key.concat(key.slice(0, nBytes - 16)); // expand key to 16/24/32 bytes
long

    var counterBlock = new Array(8);
    ctrTxt = ciphertext.slice(0, 8);
    for (var i=0; i < 8; i++) counterBlock[i] = ctrTxt.charCodeAt(i);

    var keySchedule = Aes.keyExpansion(key);

    var nBlocks = Math.ceil((ciphertext.length - 8) / blockSize);
    var ct = new Array(nBlocks);
    for (var b=0; b < nBlocks; b++) ct[b] = ciphertext.slice(8 + b * blockSize,
8 + b * blockSize + blockSize);
    ciphertext = ct;

    var plaintext = new Array(ciphertext.length);

    for (var b=0; b < nBlocks; b++) {

        for (var c=0; c < 4; c++) counterBlock[15 - c] = ((b) >>> c * 8) & 0xff;
        for (var c=0; c < 4; c++) counterBlock[15 - c - 4] = (((b + 1) / 0x100000000 - 1) >>>
c * 8) & 0xff;

        var cipherCtr = Aes.cipher(counterBlock, keySchedule);

        var plaintextByte = new Array(ciphertext[b].length);
        for (var i=0; i < ciphertext[b].length; i++) {

            plaintextByte[i] = cipherCtr[i] ^ ciphertext[b].charCodeAt(i);
            plaintextByte[i] = String.fromCharCode(plaintextByte[i]);

```

```

    }
    plaintxt[b] = plaintxtByte.join('');
  }

  var plaintext = plaintxt.join('');
  plaintext = Utf8.decode(plaintext);

  return plaintext;
}

var Base64 = {}; // Base64 namespace

Base64.code =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";

Base64.encode = function(str, utf8encode) { //
  utf8encode = (typeof utf8encode == 'undefined') ? false : utf8encode;
  var o1, o2, o3, bits, h1, h2, h3, h4, e=[], pad = '', c, plain, coded;
  var b64 = Base64.code;

  plain = utf8encode ? str.encodeUTF8() : str;

  c = plain.length % 3;   if (c > 0) { while (c++ < 3) { pad += '='; plain +=
'\0'; } }

  for (c=0; c<plain.length; c+=3) {
    o1 = plain.charCodeAt(c);
    o2 = plain.charCodeAt(c+1);
    o3 = plain.charCodeAt(c+2);

    bits = o1<<16 | o2<<8 | o3;

    h1 = bits>>18 & 0x3f;
    h2 = bits>>12 & 0x3f;
    h3 = bits>>6  & 0x3f;
    h4 = bits & 0x3f;

    e[c/3] = b64.charAt(h1) + b64.charAt(h2) + b64.charAt(h3) + b64.charAt(h4);
  }
  coded = e.join('');
  coded = coded.slice(0, coded.length-pad.length) + pad;

  return coded;
}

Base64.decode = function(str, utf8decode) {
  utf8decode = (typeof utf8decode == 'undefined') ? false : utf8decode;
  var o1, o2, o3, h1, h2, h3, h4, bits, d=[], plain, coded;
  var b64 = Base64.code;

  coded = utf8decode ? str.decodeUTF8() : str;

  for (var c=0; c<coded.length; c+=4) {
    h1 = b64.indexOf(coded.charAt(c));
    h2 = b64.indexOf(coded.charAt(c+1));
    h3 = b64.indexOf(coded.charAt(c+2));
    h4 = b64.indexOf(coded.charAt(c+3));

    bits = h1<<18 | h2<<12 | h3<<6 | h4;

```

```

    o1 = bits>>>16 & 0xff;
    o2 = bits>>>8 & 0xff;
    o3 = bits & 0xff;

    d[c/4] = String.fromCharCode(o1, o2, o3);

    if (h4 == 0x40) d[c/4] = String.fromCharCode(o1, o2);
    if (h3 == 0x40) d[c/4] = String.fromCharCode(o1);
  }
  plain = d.join('');

  return utf8decode ? plain.decodeUTF8() : plain;
}

var Utf8 = {};

Utf8.encode = function(strUni) {

  var strUtf = strUni.replace(
    /[\u0080-\u07ff]/g, // U+0080 - U+07FF => 2 bytes 110yyyyy, 10zzzzzz
    function(c) {
      var cc = c.charCodeAt(0);
      return String.fromCharCode(0xc0 | cc>>6, 0x80 | cc&0x3f); }
  );
  strUtf = strUtf.replace(
    /[\u0800-\uffff]/g, // U+0800 - U+FFFF => 3 bytes 1110xxxx, 10yyyyyy,
    10zzzzzz
    function(c) {
      var cc = c.charCodeAt(0);
      return String.fromCharCode(0xe0 | cc>>12, 0x80 | cc>>6&0x3F, 0x80 |
cc&0x3f); }
  );
  return strUtf;
}

Utf8.decode = function(strUtf) {

  var strUni = strUtf.replace(
    /[\u00e0-\u00ef][\u0080-\u00bf][\u0080-\u00bf]/g, // 3-byte chars
    function(c) {
      var cc = ((c.charCodeAt(0)&0x0f)<<12) | ((c.charCodeAt(1)&0x3f)<<6) | (
c.charCodeAt(2)&0x3f);
      return String.fromCharCode(cc); }
  );
  strUni = strUni.replace(
    /[\u00c0-\u00df][\u0080-\u00bf]/g,
    function(c) {
      var cc = (c.charCodeAt(0)&0x1f)<<6 | c.charCodeAt(1)&0x3f;
      return String.fromCharCode(cc); }
  );
  return strUni;
}

/* -----
----- */
jQuery(document).ready(function(jQuery){

$('#encrip').click(function(){

  $('#crip').show();

  var text_encrypt = Aes.Ctr.encrypt( $('#texto').val() , $('#pass').val() , 256
);

```

```
$('#crip').text(text_encrypt);  
}  
  
$('#dec_encrip').click(function(){  
    $('#dec_crip').show();  
    var text_decrypt = Aes.Ctr.decrypt($('#dec_texto').val() ,  
    $('#dec_pass').val(), 256 );  
    $('#dec_crip').text(text_decrypt);  
})  
});
```