Національний університет «Полтавська політехніка імені Юрія Кондратюка»

(повне найменування вищого навчального закладу)

\_Навчально-науковий інститут інформаційних технологій та робототехніки\_

(повна назва факультету)

\_Кафедра комп’ютерних та інформаційних технологій і систем\_

(повна назва кафедри)

**Пояснювальна записка**

**до дипломного проекту (роботи)**

|  |
| --- |
| бакалавра |

(освітньо-кваліфікаційний рівень)

на тему:

|  |
| --- |
| Розроблення сайту малого підприємства наприкладі автомийки |
|  |

Виконав: студент 4 курсу, групи 401-ТН

спеціальності

|  |
| --- |
| 122 КОМП’ЮТЕРНІ НАУКИ |

(шифр і назва напряму)

|  |
| --- |
| СКОРОДЕНКО.Д.С |

(прізвище та ініціали)

Керівник к.т.н., доцент Альошин С.П. В.

(прізвище та ініціали)

Рецензент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Полтава – 2021 року

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»**

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

**КАФЕДРА КОМП’ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І**

**СИСТЕМ**

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

**Спеціальність 122 «Комп’ютерні науки»**

**на тему**

**«**Розроблення сайту малого підприємства наприкладі автомийки**»**

**Студента групи 401-ТН Скороденко Дмитро Сергійович**

Керівник роботи:

к.т.н., доцент Альошин С.П.

Консультант:

к.т.н., доцент, Головко Г.В.

Завідувач кафедри:

к.т.н., доцент Головко Г.В.

Полтава – 2021

**РЕФЕРАТ**

Пояснювальна записка містить: 64с., 30 рисунки, 24 джерел, 1 додаток.

**Об’єкт дослідження:** інформаційні системи, що стосуються обслуговування автомобілів головних міст України та зокрема міста Полтава. Дізнатися про діяльність цих інформаційних систем.

**Предмет досліджень:** інструментальні засоби розробки інформаційної системи.

**Мета кваліфікаційної роботи:** розроблення і впровадження інформаційної системи обслуговування автомобілів м. Полтава.

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

**ABSTRACT**

The explanatory note contains: 64 pages., 30 figures, 24 sources, 1 appendix.

Object of research: information systems related to car service in the capitals of Ukraine and in particular the city of Poltava. Learn about the activities of these information systems.

**Subject of research**: tools for information system development.

**The purpose of the qualification work**: development and implementation of an information system for car maintenance in Poltava.

**Keywords**: web-site, database, database management system, personal computer, web-server, control panel.

**ЗМІСТ**

[ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ 6](#_Toc75292323)

[ВСТУП 7](#_Toc75292324)

[РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ 8](#_Toc75292325)

[1.1 Актуальність роботи 8](#_Toc75292326)

[1.2 Поняття інформаційної системи 9](#_Toc75292327)

[1.2.1 Фактори, що обумовлюють впровадження інформаційних систем 11](#_Toc75292328)

[1.2.2 Сучасний стан розвитку інформаційних систем 12](#_Toc75292329)

[1.2.3 Аналіз причин неефективної роботи інформаційних систем 14](#_Toc75292330)

[1.3 Функціональна складова автоматизованої інформаційної системи 16](#_Toc75292331)

[1.4 Огляд існуючих в Інтернеті інформаційних систем про надання послуг в обслуговуванні автомобілів в містах України 24](#_Toc75292332)

[1.5 Постановка задачі дослідження 26](#_Toc75292333)

[РОЗДІЛ 2 ПРОЕКТНІ РІШЕННЯ 28](#_Toc75292334)

[2.1 Веб-розробка і програмування 28](#_Toc75292335)

[2.2 Вибір мов і засобів програмування 29](#_Toc75292336)

[2.2.1 Вибір клієнтської мови програмування 29](#_Toc75292337)

[2.2.2 Вибір серверної мови програмування 30](#_Toc75292338)

[2.3 Вибір системи керування базами даних 31](#_Toc75292339)

[2.4 Вибір фреймворка 32](#_Toc75292340)

[2.5 Вибір середовища програмування 34](#_Toc75292341)

[2.5.1 Вибір локального сервера 35](#_Toc75292342)

[2.5.2 Вибір візуального середовища програмування 35](#_Toc75292343)

[2.5.3 Вибір програми керування БД 37](#_Toc75292344)

[РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ 39](#_Toc75292345)

[3.1 Розробка структури 39](#_Toc75292346)

[3.2 Модель схеми бази даних 39](#_Toc75292347)

[3.3 Діаграми варіантів використання інформаційної системи 42](#_Toc75292348)

[3.4 Реєстрація користувачів 44](#_Toc75292349)

[3.5 Створення головної сторінки 44](#_Toc75292350)

[3.6 Вхід до панелі управління адміністратора 47](#_Toc75292351)

[РОЗДІЛ 4 ТЕСТУВАННЯ 53](#_Toc75292352)

[4.1 Тестування ІС 53](#_Toc75292353)

[4.2 Тест-план 53](#_Toc75292354)

[ВИСНОВКИ 56](#_Toc75292355)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 57](#_Toc75292356)

[ДОДАТОК А 59](#_Toc75292357)

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ**

**АІС** – 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.

**ІС** – Інформацíйна система.

**MySQL –** вільна система керування реляційними базами даних.

## ВСТУП

В ХХІ 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 ім цікава та потрібна.

Інтернет ˗ всесвітня система взаємосполучених 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, Іне́т, Тенета, Міжмережжя, Інтернеті [[1]](https://uk.wikipedia.org/wiki/Інтернет).

## РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1 Актуальність роботи

Для виконання дипломної роботи була поставлена задача створити інформаційну, що стосуються обслуговування автомобілів головних міст України та зокрема міста Полтава. 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 [1].

Офіційний сайт має наступні розділи:

1) новинна інформація;

2) нормативні документи, положення;

3) напрямки діяльності;

4) структура закладу;

5) підготовка фахівців;

6) 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 [2].

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 [3].

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 вищий щабель [4].

### 1.2 Поняття інформаційної системи

Інформацíйна 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 [21, с. 167].

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.

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) [5].

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 [6].

1.2.1 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 [7].

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 вищий 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 [22, c. 114-122].

1.2.2 Сучасний стан розвитку інформаційних систем

Для сучасних умов характерне застосування високоефективних внутрішньофірмових систем інформації, що грунтуються 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 [8].

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 - вверх”, конкретизація інформації “зверху - вниз”, а також уніфікується інформаційний процес, спрямований на отримання науково-технічної, планової, контрольної, облікової і аналітичної інформації [9].

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:

* прямий, 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 [10].

1.2.3 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 [11];
* 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.

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 пітримуючою 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 в управлінні організацією [12].

## 1.3 Функціональна складова автоматизованої інформаційної системи

Функціональна частина АІС є домінуючою. Вона завжди пов’язана з проблемними сферами і фактично є моделлю системи управління конкретним об’єктом. До функціональної частини належать ті елементи, які визначають її функціональні можливості, а саме: призначення, виконувані управлінські функції та функції з обробки інформації. Основними елементами функціональної частини автоматизованої інформаційної системи є: функціональні підсистеми, блоки, або комплекси задач та окремі задачі. Загальна структура автоматизованої інформаційної системи (див. рис. 1.1)

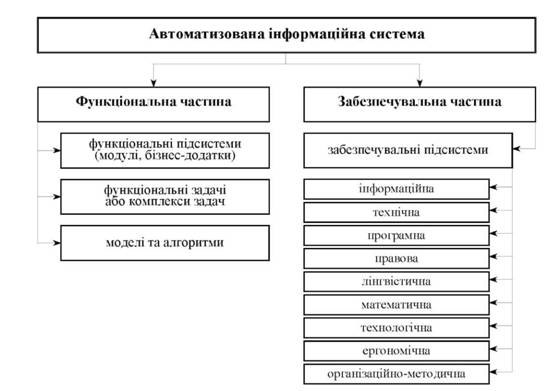


Рис. 1.1 – Загальна структура автоматизованої інформаційної системи

Функціональна підсистема - це відносно самостійна частина системи, виділена за спільністю функціональних ознак управління. В автоматизованих інформаційних системах органів казначейства функціональні підсистеми, як правило, виокремлюють за такими ознаками:

* стадіями управління (прогнозування, планування, облік, звітність, аналіз, контроль тощо);
* видами основної діяльності (доходи, видатки, трансферти тощо);
* організаційною структурою (структурні підрозділи); - функціональною ознакою (виконувані функції) [13].

Серед функціональних підсистем за ознакою управління складовими казначейської діяльності у процесі касового виконання бюджетів можна виділити: управління коштами єдиного казначейського рахунка, управління доходами державного й місцевих бюджетів, управління видатками бюджетів, управління внутрішніми казначейськими операціями, бухгалтерський облік і звітність про виконання бюджетів тощо. Незважаючи на те, що деякі функціональні підсистеми на різних об’єктах можуть мати одне й те саме найменування (наприклад, підсистема бухгалтерського обліку), їх внутрішній зміст щодо різних об’єктів значно відрізняється.

При виділенні функціональних підсистем мають бути визначені такі їх параметри:

* мета функціонування підсистеми;
* вид керованих ресурсів;
* особливості показників, що розраховуються у підсистемі; - підрозділи, які здійснюють управління.

Органи казначейства у своїй діяльності використовують автоматизовану систему виконання бюджетів "Є-Казна". Цей програмний комплекс призначений для автоматизації операцій з виконання як дохідної, так і видаткової частин бюджетів усіх рівнів, фондів соціального страхування, що проводяться органами казначейства. У частині виконання дохідної частини бюджетів АС "Є-Казна" дозволяє:

- проводити зарахування податків, зборів, інших платежів до бюджетів, пенсійного фонду та фондів соціального страхування;

- розподіляти всі надходження від платників на два потоки: доходи державногота доходи місцевих бюджетів у розрізі загального та спеціального фондів;

- розподіляти кошти згідно з чинним законодавством між різними бюджетами, фондами бюджетів, рівнями бюджетів, розпорядниками та одержувачами бюджетних коштів за допомогою спеціальних модулів системи;

- своєчасно та у повному обсязі перераховувати всі надходження за призначенням;

- формувати оперативну та достовірну звітність про виконання бюджетів за доходами на основі даних бухгалтерського обліку [14].

Відповідно до виділених функціональних підсистем та з урахуванням вимог управління визначається склад задач, що розв’язуються. У функціональних задачах, для автоматизації яких призначена підсистема, відображаються специфічні особливості кожної конкретної підсистеми певного об’єкта. Саме задача є об’єктом розробки, впровадження та експлуатації кінцевим користувачем. Зміст задачі визначає сукупність вихідних показників, які формуються й обчислюються в задачі за відповідними алгоритмами. З появою нових інформаційних технологій поняття "задача" розглядається як закінчений комплекс опрацювання інформації, що забезпечує видачу необхідної інформації для прийняття рішень управлінським персоналом.

Вибір та обґрунтування складу функціональних задач є одним з найважливіших елементів створення інформаційних систем. Розвинені автоматизовані інформаційні системи дають змогу реалізувати багатоваріантні алгоритми розрахунку показників на основі вибору з банку моделей найефективніших математичних методів, моделей й алгоритмів для конкретного об’єкта управління. При цьому кожна конкретна задача при розробці автоматизованих інформаційних систем повинна розглядатися в інформаційному зв’язку з іншими задачами даної підсистеми та інших підсистем, а також із зовнішніми автоматизованими інформаційними системами.

Зважаючи на складність та комплексний характер функціональних задач вибір їх складу повинен здійснюватися з урахуванням основних фаз управління: прогнозування, планування, обліку, контролю, аналізу та регулювання.

Прогнозування - управлінська функція, за допомогою якої здійснюється формування й наукове обґрунтування перспектив розвитку об’єкта управління, можливих змін його стану у майбутньому.

За допомогою автоматизованих інформаційних систем розв’язуються комплекси задач із прогнозування різних показників у сфері бюджетної діяльності.

Планування - функція, за допомогою якої реалізується мета управління. Втілення цієї функції забезпечується формуванням планів, згідно з якими буде організовано функціонування об’єкта управління.

Облік, контроль і аналіз - функції, що забезпечують одержання інформації про стан керованої системи за певний проміжок часу; установлення фактичного стану об’єкта управління; визначення відхилень, що виходять за межі допустимих даних; встановлення причин відхилення фактичних показників від запланованих. Комплекси задач, що розв’язуються на цих фазах управління процесом виконання бюджету, виконуються на різних рівнях управління, мають різний період вирішення та обсяги інформації, що опрацьовується. Облік ведеться на нижчому й середньому рівнях. На цю фазу управління припадає найбільше інформаційне навантаження. Аналіз проводиться на середньому й вищому рівнях управління. Ця фаза управління оперує агрегованою інформацією з тривалим періодом вирішення задач. Контроль здійснюється на всіх рівнях управління, залучаючи значний обсяг інформації з тривалим періодом її опрацювання.

Регулювання - корегування діяльності об’єкта управління за наявності відхилень фактичних показників від запланованих; прийняття рішень для виправлення ситуації, що виникла; виведення об’єкта управління на заплановані показники. Для формалізованого опису задач регулювання використовують методи й моделі календарного та мережного планування, транспортні моделі, а також моделі оперативного управління. Підсумковою інформацією цієї фази є календарні й мережні графіки корегування процесу виконання бюджету. Регулювання здійснюється на середньому та вищому рівнях управління.

Сучасні автоматизовані інформаційні системи здатні обробляти інформацію за всіма функціями управлінської діяльності та передавати її на всі рівні управління. Особливо ефективно в автоматизованих інформаційних системах реалізуються функції контролю, аналізу й регулювання, оскільки вся вхідна інформація для розв’язання цих функціональних задач сформована ще під час розв’язання задач планування й обліку і знаходиться у базі даних. Загалом автоматизована інформаційна система має забезпечити дієвий контроль стану об’єкта управління й оперативне прийняття обґрунтованих рішень.

В умовах змінного зовнішнього середовища АІС повинна мати гнучку структуру і бути відкритою для внесення необхідних змін у розроблену модель, забезпечувати нарощування функціональних можливостей по мірі необхідності. При цьому функціональна структура АІС має орієнтуватися на ті інформаційні потреби кінцевих користувачів, які змінюються в умовах ринку, та відображати зміст і специфіку функцій управління конкретним економічним об’єктом.

Найбільшою мірою функціональність АІС забезпечується за допомогою принципу модульності. Модуль являє собою цілісну групу елементів системи, яка описана тільки своїми входами і виходами. Модуль як частина системи відображає зв’язок між елементами, тобто обмін інформацією.

У різних автоматизованих інформаційних системах як модуль можуть розглядатися комплекси задач, АРМи або функціональна підсистема. Кожний прикладний модуль системи має обслуговувати деяку інформаційну сферу. Головною вимогою при розробці модулів повинна бути орієнтація системи на автоматизацію управління діяльністю об’єкта, а не на розв’язання локальних функціональних задач. Завдяки інтеграції модулів у єдину систему забезпечується комплексність системи.

Модульний підхід дає змогу значно спростити опис системи та зробити видимими й доступними для розуміння найскладніші системи. За допомогою принципу модульності вирішується проблема розподілу задач між учасниками процесу управління. Це пов’язано з тим, що деякі задачі можуть бути повністю розв’язані на одному робочому місці, а інші потребують участі багатьох управлінських працівників.

Функціональна частина автоматизованої інформаційної системи на практиці може реалізуватися на різній технічній базі та із застосуванням різних технологій обробки даних (централізованих або децентралізованих).

Централізована технологія передбачає виконання функціональних завдань фахівцями інформаційних систем у повному обсязі безпосередньо в обчислювальному центрі (ОЦ).

Децентралізована технологія реалізується через використання персональних комп’ютерів, зосереджених на окремих робочих місцях управлінського персоналу, так звані, автоматизовані робочі місця (АРМ), на яких розв’язуються комплекси функціональних задач [15].

АРМ управлінського персоналу - це програмно-технічний комплекс, винесений на робоче місце конкретного управлінського працівника, що дає змогу в автоматизованому режимі виконувати покладені на нього функціональні задачі управління в інтерактивному режимі "користувач-ПЕОМ". Основним принципом функціонування АРМ є орієнтація на розв’язання певного класу задач, об’єднаних загальною технологією обробки інформації, єдністю режимів роботи й експлуатації.

АРМ призначене для розв’язання комплексу управлінських задач фахівцем або групою фахівців на різних рівнях управлінської діяльності. Тільки в умовах функціонування АРМів управлінського персоналу можуть бути автоматизовані всі елементи процесу управління, можливе зближення процесів збору вхідної інформації, формування, відображення та використання вихідної інформації. При цьому управлінський персонал як користувач інформаційних систем виступає у ролі оператора ПЕОМ, тобто здійснює введення інформації у систему, підтримує її в актуальному стані, обробляє інформацію і використовує здобуті результати в управлінні. Таким чином, він продовжує здійснювати традиційні контрольно-аналітичні функції прийняття управлінських рішень, але на вищому автоматизованому рівні. При цьому у кожній персональній інформаційній системі має бути забезпечена можливість для користувача виконувати такі дії:

* обробляти текстові документи;
* здійснювати пошук необхідної інформації у довідкових фондах та архівах користувача;
* обмінюватися документами й даними;
* узгоджувати, уточнювати, модернізувати задачі управління;
* приймати рішення в умовах невизначеності або недостатності інформації.

Виконання зазначених дій можливе за умови роботи конкретного робочого місця у багатофункціональному режимі. Спільне функціонування ПЕОМ у процесі колективного опрацювання даних досягається за допомогою обладнання АРМ спеціальними виходами через системний пристрій і каналами зв’язку в ПЕОМ і баз даних (БД) інших користувачів.

Децентралізована обробка даних з використанням АРМ є ефективнішою за централізовану, оскільки забезпечує прямий доступ користувача-фахівця до ЕОМ без посередників (операторів, програмістів), які можуть вплинути на правдивість оброблюваної інформації, а також знизити оперативність виконання відповідних завдань.

За децентралізованого використання обчислювальної техніки автоматизована інформаційна система може створюватися як:

1) сукупність автономних систем, об’єднаних лініями зв’язку в локальну мережу, так звана, однорівнева система;

2) дворівнева система, в якій на автоматизованих робочих місцях (АРМах) здійснюється попередня обробка первинної інформації з подальшою її передачею до інформаційно-обчислювального центру (ІОЦ) для остаточного завершення процесу обробки інформації.

Функціонування однорівневої системи децентралізованої обробки даних забезпечується шляхом встановлення на окремих робочих місцях персональних комп’ютерів. Організація дворівневої системи обробки даних передбачає обладнання великої електронно-обчислювальної машини з виносними пультами і встановлення їх на робочих місцях фахівців.

На практиці перевагу надають дворівневій системі децентралізованої обробки даних, оскільки вона поєднує переваги як децентралізованої системи обробки інформації, у яких технічні засоби наближені до місць її виникнення, так і централізованих систем, коли об’ємна і складна робота з розв’язання функціональних задач виконується в інформаційно-обчислювальних центрах.

При цьому як в однорівневій, так і у дворівневій системах децентралізованої обробки даних функціональними елементами автоматизованих систем є функціонально-спеціалізовані АРМ (ФСАРМ). Враховуючи зазначене під АРМ слід розуміти набір персональної комп’ютерної техніки, яка через залучення обчислювальних потужностей великої ЕОМ дає змогу виконувати на робочому місці інформаційне обслуговування користувача-фахівця в обсязі та режимі, які необхідні для виконання ним своїх службових функцій.

Сукупність функціонально-спеціалізованих АРМ складає основу функціональної частини автоматизованої інформаційної системи органів казначейства. їх кількість і структура в АІС на різних рівнях казначейської системи залежить від обраної технології опрацювання даних та поділу функцій між виконавцями. Основними чинниками, які впливають на формування функціонально-спеціалізованих АРМ є:

- організаційна структура органу ;

- специфікація операцій;

- спеціалізація окремих груп працівників і структурних підрозділів, розмежування функціональних повноважень між ними [16].

**1.4 Огляд існуючих в Інтернеті інформаційних систем про надання послуг в обслуговуванні 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 авто без проблем.

Зараз розглянемо приклади деяких сайтів, що стосуються обслуговування автомобілів:

1. «Auto Rent Center» Обслуговування автомобілів в місті Харків autorent-center.com представлений на рис. 1.1

Недоліки представленого сайту:

* відсутня можливість бронювання на сайті;
* немає представлених зображень послуг, які надає компанія;

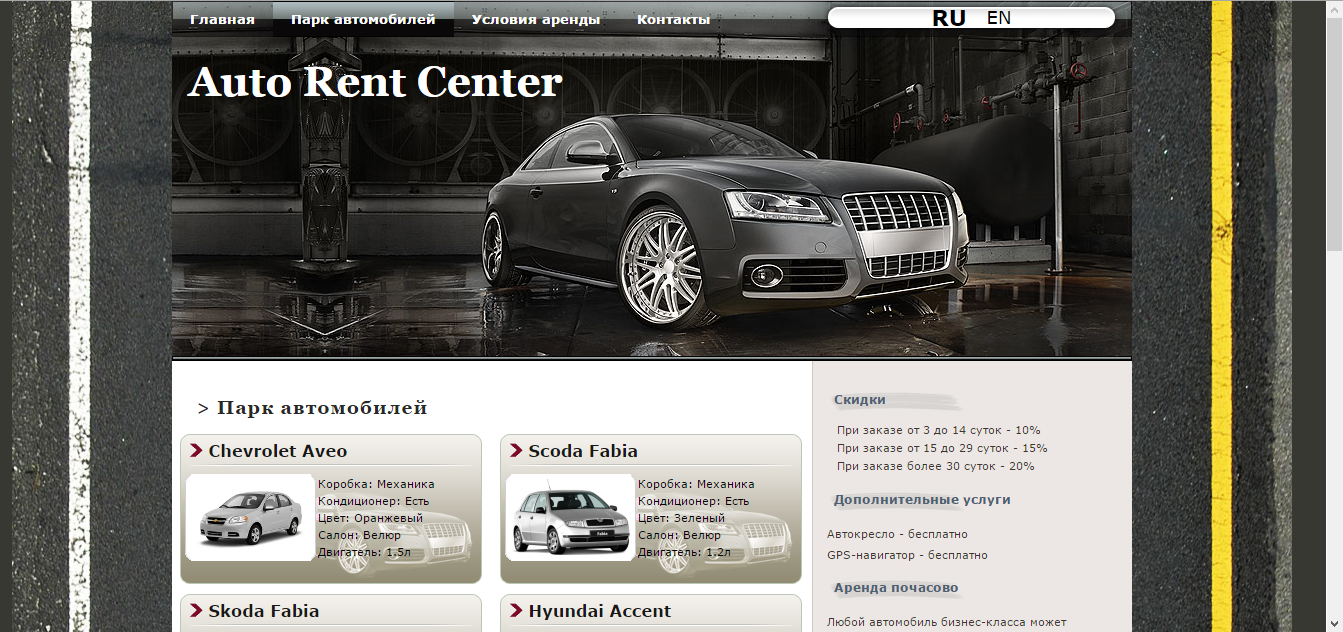


Рисунок 1.1 − Обслуговування автомобілів у Харкові autorent-center.com

2. “Smartauto” – автомийка в місті Київ

Недоліки представленого сайту:

* Не оптимізований дизайн

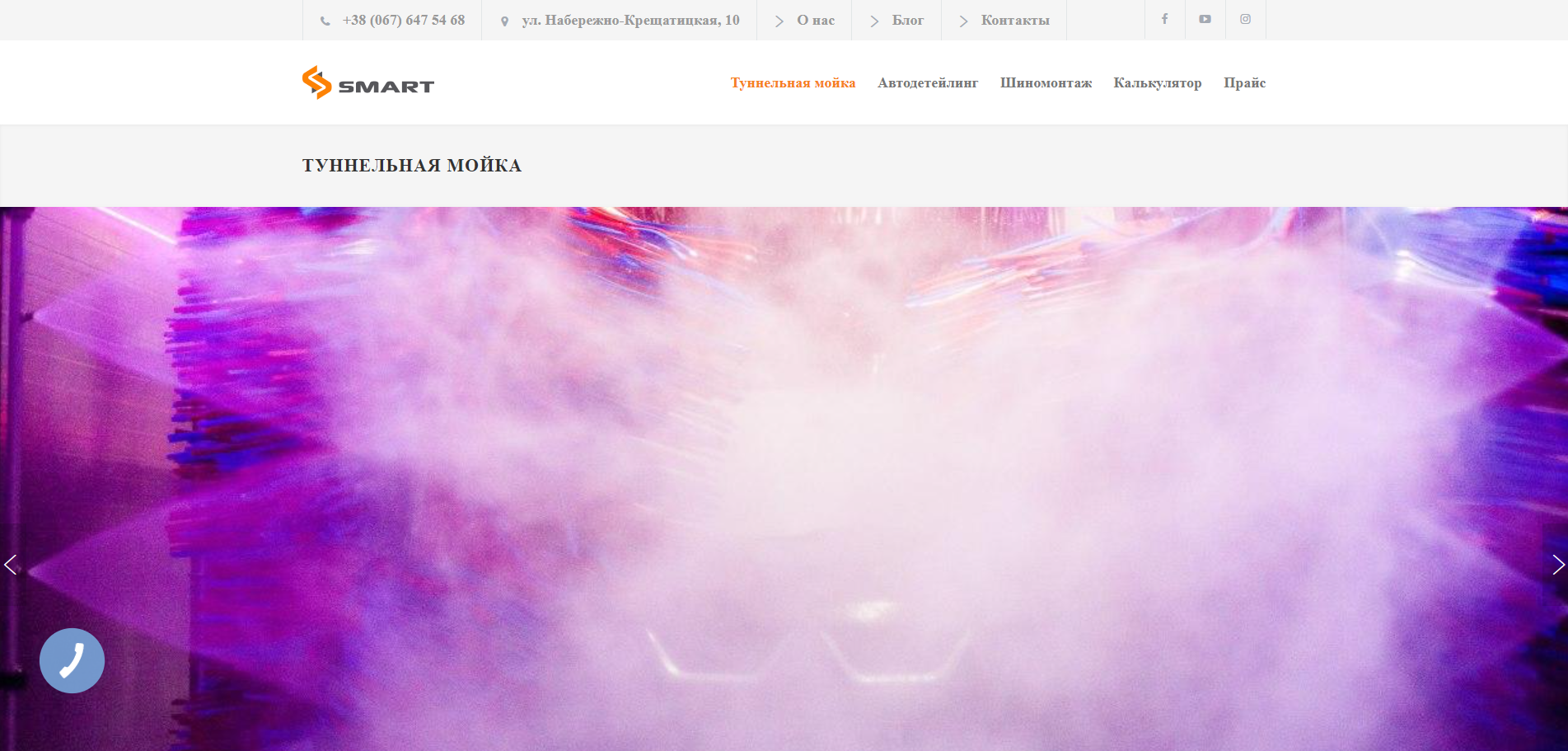


Рисунок 1.2 − “Smartauto” – автомийка в місті Київ

3 “Garage detailinglab” – автомийка в місті Київ

Недоліки представленого сайту:

* Недоліків не знайдено

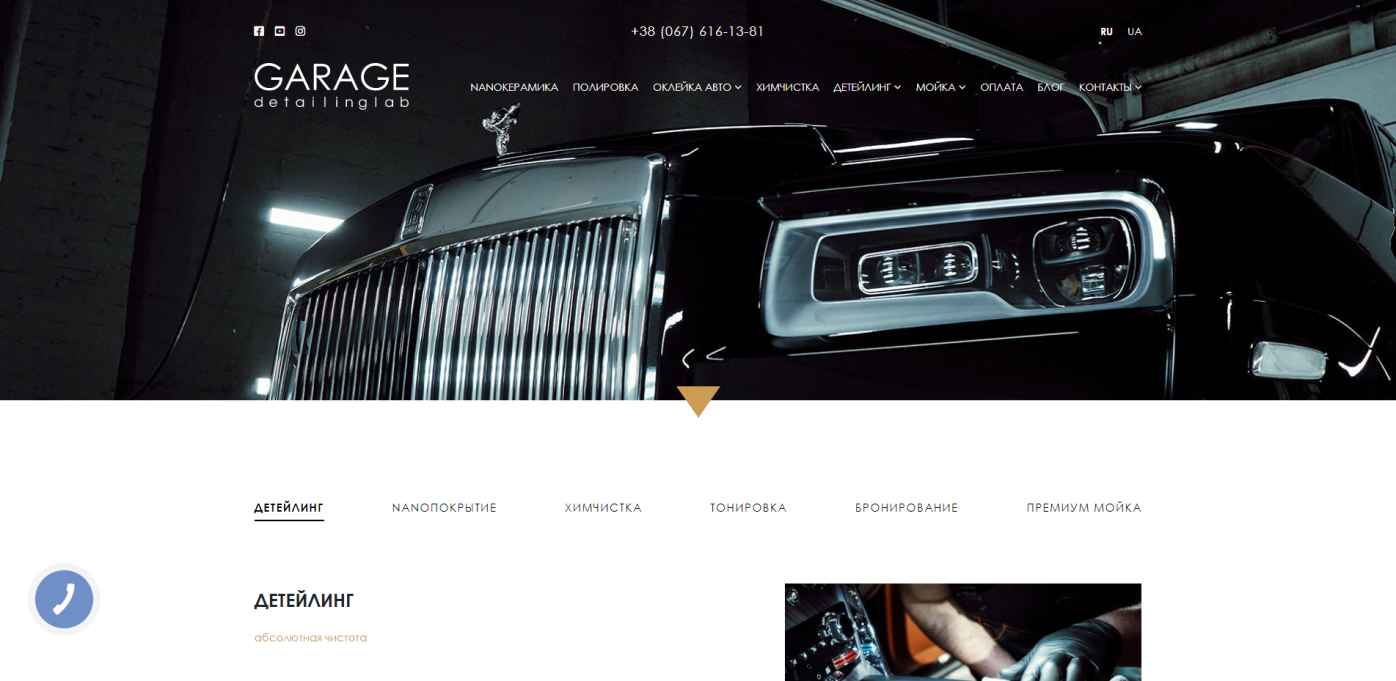


Рисунок 1.3 − “Garage detailinglab” – автомийка в місті Київ

1. «Rent-a-car» Обслуговування автомобілів у місті Дніпро.

rentacar-dnepropetrovsk.com.ua представлений на рис. 1.3

Недоліки представленого сайту:

* застарілий дизайн;
* відсутня класифікація автомобілів [17].



Рисунок 1.4 − Обслуговування автомобілів в місті Дніпро

rentacar-dnepropetrovsk.com.ua

**1.5 Постановка задачі дослідження**

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

* бронювання місць для авто;
* можливість детального перегляду послуг;
* класифікація послуг;
* детальна інформація щодо умов замовлення послуг;
* можливість додавання нових послуг для автомобілів на сайт адміністрацією;
* повідомлення адміністрацією про діючі акції;
* зв’язок з довідковою службою, зручним для користувача, шляхом (телефон, електронна пошта, skype) [18].

## РОЗДІЛ 2 ПРОЕКТНІ РІШЕННЯ

**2.1 Веб-розробка і програмування**

**Веб-розробка** – процес створення веб-сайтів або веб-додатків. Основними етапами процесу є веб-дизайн, верстка сторінок, програмування для веб на стороні клієнта і сервера, а також конфігурування веб-сервера.

На сьогоднішній день існують кілька етапів розробки [веб-сайта](http://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82):

* [Проектування](http://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) сайту або веб-додатки (збір і аналіз вимог розробка [Технічного завдання](http://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D1%85%D0%BD%D1%96%D1%87%D0%BD%D0%B5_%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F), проектування [Інтерфейс користувача](http://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BA%D0%BE%D1%80%D0%B8%D1%81%D1%82%D1%83%D0%B2%D0%B0%D1%87%D0%B0));
* Розробка [креативної](http://uk.wikipedia.org/wiki/%D0%9A%D1%80%D0%B5%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D1%96%D1%81%D1%82%D1%8C) концепції сайту;
* Створення [дизайн-концепції](http://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD) сайту;
* Створення [макетів](http://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D0%BA%D0%B5%D1%82) сторінок;
* Створення [мультимедіа](http://uk.wikipedia.org/wiki/%D0%9C%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BC%D0%B5%D0%B4%D1%96%D0%B0) та [FLASH](http://uk.wikipedia.org/wiki/Adobe_Flash)-елементів;
* [Верстка](http://uk.wikipedia.org/wiki/%D0%92%D0%B5%D1%80%D1%81%D1%82%D0%BA%D0%B0) сторінок і [дизайнів](http://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD);
* [Програмування](http://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) (розробка функціональних інструментів) або інтеграція в систему управління вмістом ([CMS](http://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%96%D0%BD%D0%BD%D1%8F_%D0%B2%D0%BC%D1%96%D1%81%D1%82%D0%BE%D0%BC));
* [Оптимізація](http://uk.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%B8%D0%BC%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%8F_%D1%81%D0%B0%D0%B9%D1%82%D1%83) і [розміщення](http://uk.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3) матеріалів сайту;
* [Тестування](http://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F) та внесення коригувань;
* Відкриття проекту на [хостингу](http://uk.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3);
* Обслуговування працюючого сайту або його програмної основи.

В залежності від поточного завдання якісь з етапів можуть бути відсутні або бути тісно пов’язані один з іншим.

**Веб-програмування** – галузь веб-розробки і різновид дизайну, в завдання якої входить проектування користувальницьких веб-інтерфейсів для сайтів або веб-додатків. Веб-дизайнери проектують логічну структуру веб-сторінок, продумують найбільш зручні рішення подачі інформації, а також займаються художнім оформленням веб-проекту. В результаті перетину двох галузей людської діяльності грамотний веб-дизайнер повинен бути знайомий з останніми веб-технологіями і володіти відповідними художніми якостями [19].

Розробка сайтів – це не зовсім програмування. Під «програмуванням» розуміють реалізацію визначеного функціоналу. При цьому програміст, як правило, не дуже задумується над інтерфейсом своєї програми (навіть якщо пише цю програму самостійно).

Сайт повинен хоча б чимось, але все ж таки виділятися від інших ресурсів мережі, інакше його, як мінімум, не запам’ятають. Тому від розробника вимагаються знання як HTML, CSS, PHP, так і Photoshop та інших програм для створення дизайну.

**2.2 Вибір мов і засобів програмування**

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.

2.2.1 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.

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.

2.2.2 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.2.1). 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.

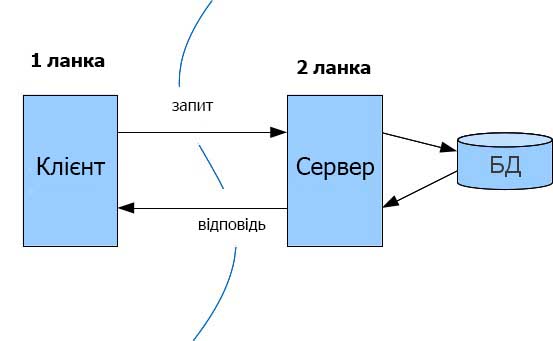


Рисунок 2.1 − Мережева організація

Раціонально 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 [20].

**2.3 Вибір системи керування базами даних**

База даних – впорядкований набір логічно взаємопов’язаних даних, що використовуються спільно та призначені для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система керування БД. Головне завдання БД – гарантоване збереження значних обсягів інформації (так звані записи даних) та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином, БД складається з двох частин: збереженої інформації та системи керування нею. З метою забезпечення ефективності доступу записи даних організовують як множину фактів (елемент даних).

Існує кілька систем управління базами даних зокрема: Microsoft Access,DB2, Informix, Oracle, SQL Server, Дінай, MySQL, PostgreSQL, Для веб-розробки доцільно використовувати MySQL оскільки це СУБД з відкритим кодом. MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система управління базами даних (СУБД) 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.

**2.4 Вибір фреймворка**

Фреймворк – програмна платформа, яка визначає структуру програмної системи, програмне забезпечення, що полегшує розробку і об’єднання різних компонентів великого програмного проекту. Вживається також слово «каркас», а деякі автори використовують його в якості основного, в тому числі не базуючись взагалі на англомовному аналогу. Можна також говорити про каркасний підхід як про підхід до побудови програм, де будь-яка конфігурація програми будується з двох частин: перша, постійна частина – каркас, незмінний від конфігурації до конфігурації і несе в собі гнізда, в яких розміщується друга, змінна частина – змінні модулі (або точки розширення).

Існує кілька веб-фреймворків реалізованих на PHP це:

* Zend Framework;
* CakePHP;
* Code Igniter;
* Kohana;
* Symfony;
* Yii;
* Laravel.

Вимоги до фреймворка такі:

* легкість створення нового проекту, читання та пошук документації;
* продуманість об’єктної моделі, можливість розширення;
* споживання пам’яті;
* швидкодія;
* зручність встановлення і налаштування.

Проаналізувавши недоліки і переваги вище перелічених феймворків, а також вимоги до них був вибраний Laravel.

Laravel – безкоштовний веб-фреймворк з відкритим кодом, призначений для розробки з використанням архітектурної моделі MVC (англ. Model View Controller – модель-представлення-контролер). Laravel випущений під ліцензією MIT. Вихідний код проекту розміщується на GitHub.

В результаті опитування sitepoint.com в грудні 2018 про найпопулярніші PHP-фреймворки, Laravel зайняв місце самого багатообіцяючого проекту на 2019 рік.

Переваги Laravel:

* Пакети (англ. Packages) – дозволяють створювати і підключати модулі в форматі Composer до додатка на Laravel. Багато додаткових можливостей вже доступні у вигляді таких модулів;
* Eloquent ORM – реалізація шаблону проектування ActiveRecord на PHP. Дозволяє строго визначити відносини між об’єктами бази даних. Стандартний для Laravel будівник запитів Fluent підтримується ядром Eloquent;
* Логіка додатка - частина розроблювального додатка, оголошена або за допомогою контролерів, або маршрутів (функцій-замикань). Синтаксис оголошень схожий на синтаксис, використовуваний в каркасі Sinatra;
* Зворотня маршрутизація пов’язує між собою посилання і маршрути що генеруються додатком, дозволяючи їх змінювати з автоматичним оновленням пов’язаних посилань. При створенні посилань за допомогою іменованих маршрутів Laravel автоматично генерує кінцеві URL;
* REST-контролери - додатковий шар для розділення логіки обробки GET- і POST-запитів HTTP;
* Автозавантаження класів - механізм автоматичного завантаження класів PHP без необхідності підключення файлів їх визначень у include. Завантаження на вимогу запобігає завантаженню непотрібних компонентів; завантажуються тільки ті з них, які дійсно використовуються;
* Укладачі уявлень (англ. View composers) - блоки коду, які виконуються при генерації подання (шаблону).
* Міграції - система управління версіями для баз даних. Дозволяє пов’язувати зміни в коді програми зі змінами, які потрібно внести в структуру БД, що спрощує розгортання і оновлення програми.
* Модульне тестування (юніт-тести) - відіграє дуже велику роль в Laravel, який сам по собі містить велику кількість тестів для запобігання регресій (помилок внаслідок поновлення коду або виправлення інших помилок).
* Сторінкове виведення (англ. Pagination) - спрощує генерацію сторінок, замінюючи різні способи вирішення цього завдання єдиним механізмом, вбудованим в Laravel [23, c. 256].

**2.5 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 сирцевого коду, компілятора абоінтерпретатора тазасобів автоматизації збірки. Іноді сюди також входятьсистеми контролю версій, засоби дляпрофілювання, 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.

Для зручної розробки веб-проекту знадобляться такі засоби програмування:

* локальний сервер;
* візуальне середовище програмування;
* програма керування БД.

2.5.1 Вибір локального сервера

Після детального аналізу локальних серверів був вибраний Open Server.

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

2.5.2 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.

Візуальним середовищем програмування був вибраний PhpStorm (рис. 2.2). PhpStorm являє собою інтелектуальний редактор для PHP, HTML і JavaScript з можливостями аналізу коду на льоту, запобігання помилок у сирцевому коді і автоматизованими засобами рефакторинга для PHP і JavaScript. Автодоповнення коду в PhpStorm підтримує специфікацію PHP 5.3, 5.4 та 5.5, включаючи генератори, співпрограми, простори імен, замикання, типажі і синтаксис коротких масивів. Присутній повноцінний SQL-редактор з можливістю редагування отриманих результатів запитів.

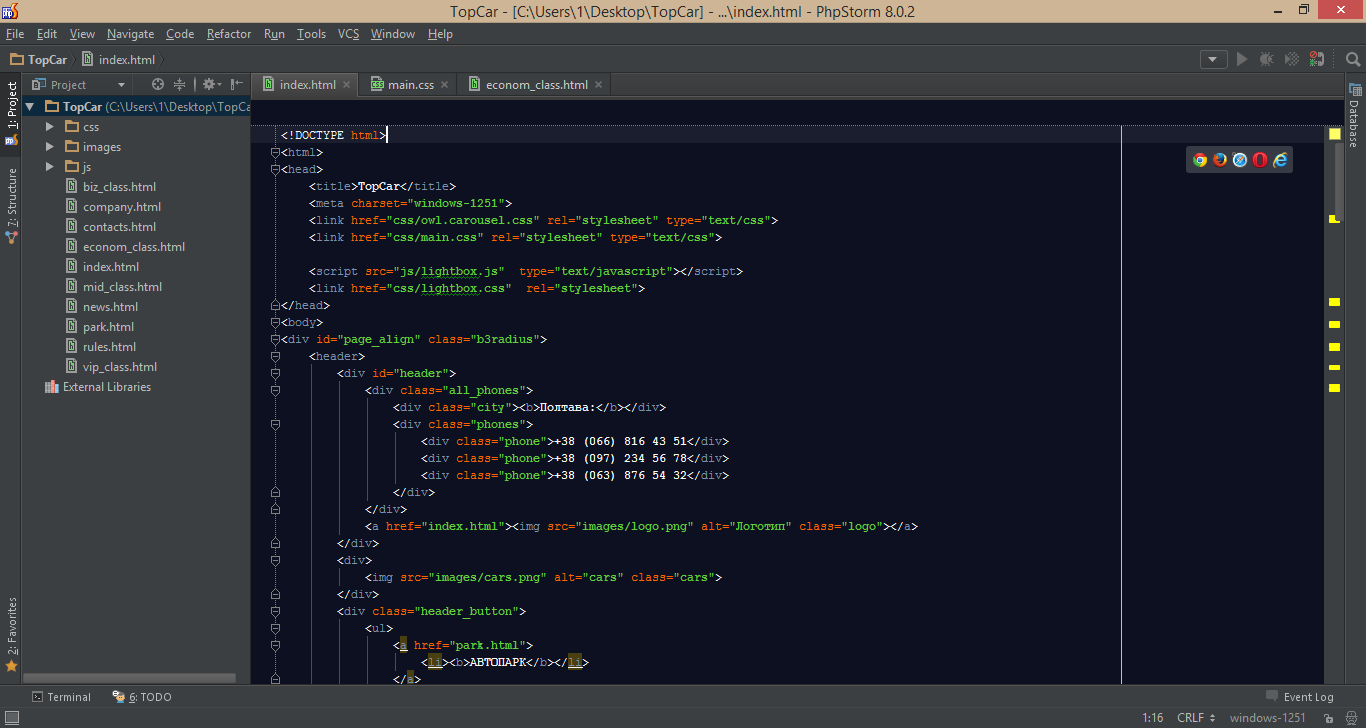


Рисунок 2.2 – Візуальне середовище програмування PhpStorm.

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

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

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

- Докладний перегляд логів всіх компонентів в реальному часі;

- Вибір HTTP, СУБД і PHP модулів в будь-якому поєднанні;

- Підтримка SSL і кириличних доменів з коробки;

- Підтримка алиасов або по іншому доменних покажчиків, а так само зручна форма їх налаштування (привіт любителям мультісайтінга в Drupal!);

- Створення локального піддомена без втрати видимості основного домену в мережі інтернет;

- Доступ до доменів (в один клік) і швидкий доступ до шаблонів конфігурації модулів;

- Багатомовний інтерфейс (Російська, Українська, Білоруська, Англійська);

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

2.5.3 Вибір програми керування БД

У середовищі програмування яке надає Open Server присутня стандартна програма для роботою з базою даних PhpMyAdmin але оскільки вона працює через браузер то користуватися нею не завжди зручно, тому для цього була вибрана десктопна програма Navicat Premium (рис. 2.3).

Navicat Premium – програма керування [базами даних](http://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) для [MySQL](http://uk.wikipedia.org/wiki/MySQL), [Oracle](http://uk.wikipedia.org/wiki/Oracle_Database), [SQLite](http://uk.wikipedia.org/wiki/SQLite), [PostgreSQL](http://uk.wikipedia.org/wiki/PostgreSQL) і [Microsoft SQL Server](http://uk.wikipedia.org/wiki/Microsoft_SQL_Server). Вона має Explorer – подібний графічний користувальницький [інтерфейс](http://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) і підтримує кілька з’єднань з базою даних для локальних і віддалених баз. Дизайн програми виконаний з урахуванням потреб найрізноманітніших аудиторій: від адміністраторів баз даних і програмістів до різноманітних підприємств/компаній, які обслуговують клієнтів і обмінюються інформацією з партнерами [24, c. 192].

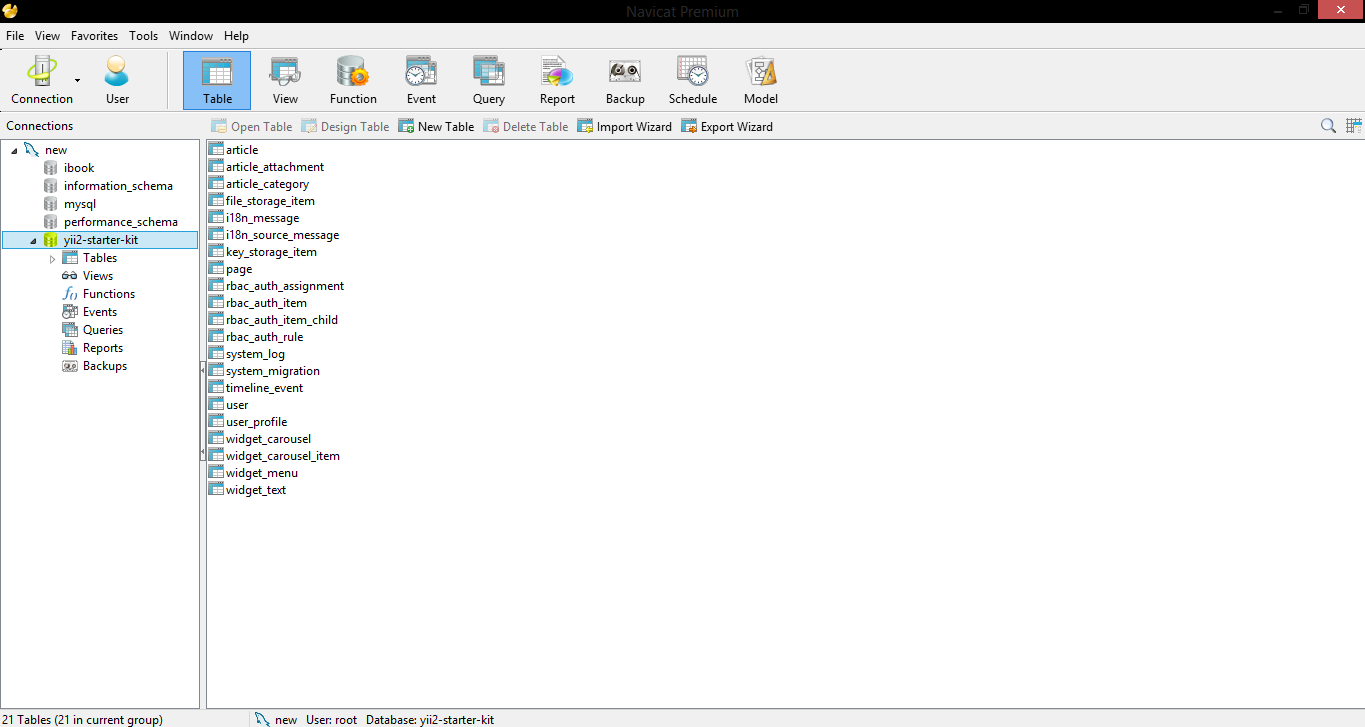


Рисунок 2.3 − програма керування БД Navicat Premium.

## 

**РОЗДІЛ 3  
ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ**

**3.1 Розробка структури**

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

* головна сторінка;
* класифікація послуг;
* бронювання місць для авто;
* панель управління адміністратора;

В свою чергу панель управління адміністратора має наступну структуру:

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

Горизонтальне навігаційне меню сайту містить 5 основних пунктів,це: Автопарк, Новости, Условия проката, О компании, Контакты. Головна – здійснюється перехід на головну сторінку, на якій знаходиться навігаційна карта.

**3.2 Модель схеми бази даних**

Інфологічна модель схеми бази даних – це рівень, який  являє собою інформаційно-логічну модель (ІЛМ) предметної області, в якій виключена надмірність даних і відображені інформаційні особливості об’єкту управління, без урахування особливостей і специфіки конкретної СУБД.  
Мета інфологічного проектування – створити структуровану інформаційну модель ПЗ, для якої розроблятиметься БД. Під час проектування на інфологічному рівні створюється інформаційно-логічна модель, яка має відповідати таким вимогам:

* коректність схеми БД, тобто адекватне відображення модельо­ваного програмного забезпечення;
* простота і зручність використання на наступних етапах проектування, тобто інфологічна модель має легко відображатися в моделі БД, що підтримується відомими СУБД (сіткові, ієрархічні, реляційні);
* модель має бути описана мовою, зрозумілою проектувальникам БД, програмістам, адміністратору і майбутнім користувачам АБ.

Основною складовою інфологічної моделі є *атрибути*, які потрібно проаналізувати і деяким чином згрупувати для подальшого зберігання в БД. Сутність інфологічного моделювання полягає у виокремленні інформаційних об’єктів ПО (таблиць), які підлягають зберіганню в БД, а також визначенні характеристик об’єктів і зв’язків між ними. Характеристиками чи властивостями об’єктів є атрибути.

Існує два підходи до інфологічного проектування: аналіз об’єктів і синтез атрибутів. Підхід, що базується на аналізі об’єктів, називається низхідним, а на синтезі атрибутів – висхідним. Економіко-організаційні системи найчастіше створюються з використанням висхідного підходу, який полягає в аналізі атрибутів і синтезі на їх основі інформаційних об’єктів.

Основними складовими елементами інфологічної моделі є інформаційний об’єкт, атрибут, запит, запитальний зв’язок, структурний зв’язок.

Інформаційний об’єкт – деяка сутність ПЗ, яку необхідно відображувати в БД з точки зору прикладної програми чи користувача БД. Це може бути предмет, факт, дія, явище чи поняття. Кожен об’єкт описується його властивостями, тобто його атрибутами. Крім атрибутів та інформаційних об’єктів, складовими частинами інфологічної моделі є інформаційний запит, запитувальний зв’язок і структурний зв’язок.

Інформаційний запит – словесний опис інформаційної потреби користувача чи прикладної програми.

Запитувальний зв’язок будується на основі запиту. Він являє собою структурований опис інформаційного запиту, в якому відображено необхідні для його реалізації об’єкти з урахуванням навігації (шляхів інформаційного пошуку) між ними.

Структурний зв’язок – асоціації, що описують ієрархічні зв’язки між парами інформаційних об’єктів, один з яких виступає як власник, а інший як підпорядкований об’єкт.

Екземпляр структурного зв’язку – це екземпляр об’єкта власника та певна сукупність зв’язаних з ним екземплярів підпорядкованого об’єкта.

Даталогічний (логічний, концептуальний) рівень формується з урахуванням специфіки і особливостей конкретної СКБД. На цьому рівні будується концептуальна модель даних, тобто спеціальним способом структурована модель ПО, яка відповідає особливостям і обмеженням вибраної СКБД. Модель логічного рівня, яка підтримується засобами конкретної СКБД, іноді називають даталогічною. Серед типів моделей, які підтримуються засобами СКБД, є ієрархічні, сіткові і реляційні та об’єктно-орієнтовані моделі баз даних.

Внутрішній рівень пов’язаний з фізичним розміщенням даних у пам’яті ЕОМ. На цьому рівні формується фізична модель БД, яка містить структури зберігання даних у пам’яті ЕОМ, включаючи опис форматів даних, порядок їх логічного чи фізичного упорядкування, розміщення за типами пристроїв, а також характеристики і шляхи доступу до даних. Від параметрів фізичної моделі залежать такі характеристики функціонування БД: обсяг пам’яті і час реакції системи. Фізичні параметри БД можна змінювати у процесі її експлуатації (не змінюючи при цьому опису інших рівнів) з метою підвищення ефективності функціонування системи. Структура таблиць-сутностей БД визначається на етапах інфологічного і логічного проектування, а формування структури — на етапі фізичного проектування БД. Структура ж таблиці – це пойменована сукупність логічно взаємозв’язаних атрибутів.

Схема бази даних системи управління контентом наведена на рис. 3.1.



Рисунок 3.1 − Схема бази даних

**3.3 Діаграми варіантів використання інформаційної системи**

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, асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

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

Діаграми варіантів використання сайту наведені на рис. 3.2.

До базових елементів розглянутої діаграми відносяться варіант використання, актор і інтерфейс.

Варіант використання застосовується для специфікації загальних особливостей поведінки системи або іншої сутності без розгляду її внутрішньої структури (наприклад, оформлення замовлення на купівлю товару, отримання інформації про кредитоспроможність клієнта, відображен­ня графічної форми на екрані монітора).

Актор – це зовнішня по відношенню до моделюється системі сутність, яка взаємодіє з системою і використовує її функціональні можливості для вирішення певних завдань. При цьому актори служать для позначення узгодженого безлічі ролей, які можуть відігравати користувачі в процесі взаємодії з проектованою системою. Ім’я актора має бути достатньо інформативним з точки зору семантики, наприклад клієнт банку, продавець магазину, пасажир авіарейсу, водій автомобіля, стільниковий телефон.

В даному випадку в системі визначені такі актори:

* Користувач, який може забронювати автомобіль та переглядати існуючу інформацію на сайті;
* Адміністратор, який володіє більш широкими правами.

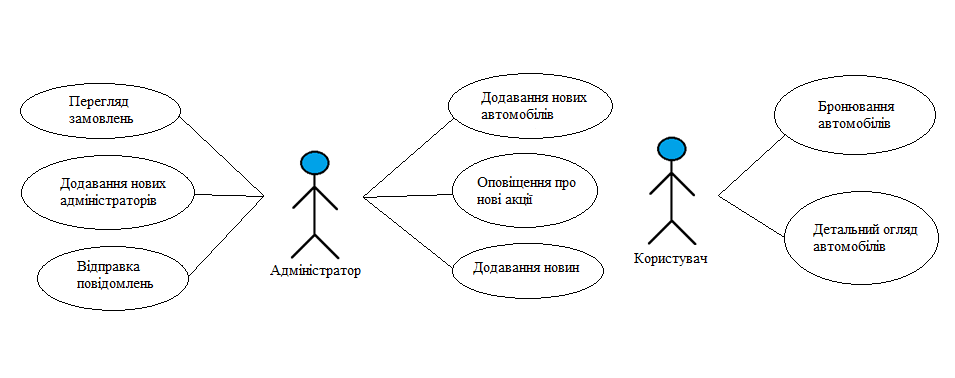


Рисунок 3.2 − Діаграми варіантів використання сайту

Так як в загальному випадку актор завжди знаходиться поза системою, його внутрішня структура ніяк не визначається. Для актора має значення тільки його зовнішнє уявлення, тобто то, як він сприймається з боку системи. Актори взаємодіють з системою за допомогою передачі і прийому повідомлень від варіантів використання. Повідомлення являє собою запит актором сервісу від системи і отримання цього сервісу. Ця взаємодія може бути виражене за допомогою асоціацій між окремими акторами і варіантами використання або класами. Крім цього, з акторами можуть бути пов’язані інтерфейси, які визначають, яким чином інші елементи моделі взаємодіють з цими акторами.

Інтерфейс служить для специфікації параметрів моделі, які видимі ззовні без вказівки їх внутрішньої структури. У діаграмах варіантів використання інтерфейси визначають сукупність операцій, які забезпечують необхідний набір сервісів або функціональності для акторів. Інтерфейси не можуть містити ні атрибутів, ні станів, ні направлених асоціацій. Вони містять лише операції без вказівки особливостей їх реалізації. Формально інтерфейс еквівалентний абстрактному класу без атрибутів і методів з наявністю тільки абстрактних операцій.

**3.4 Реєстрація користувачів**

Реєстрації та авторизації користувачів на сайті немає, тому що я вважаю, це недоцільно, адже всі необхідні дані замовник буде надавати при отриманні послуг для авто. Для бронювання через сайт достатньо заповнити вказані поля: П.І.П., E-mail, телефон, адресу та по бажанню написати подробиці чи побажання, що стосуються замовлення.

**3.5 Створення головної сторінки**

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. 3.3 та рис. 3.4).

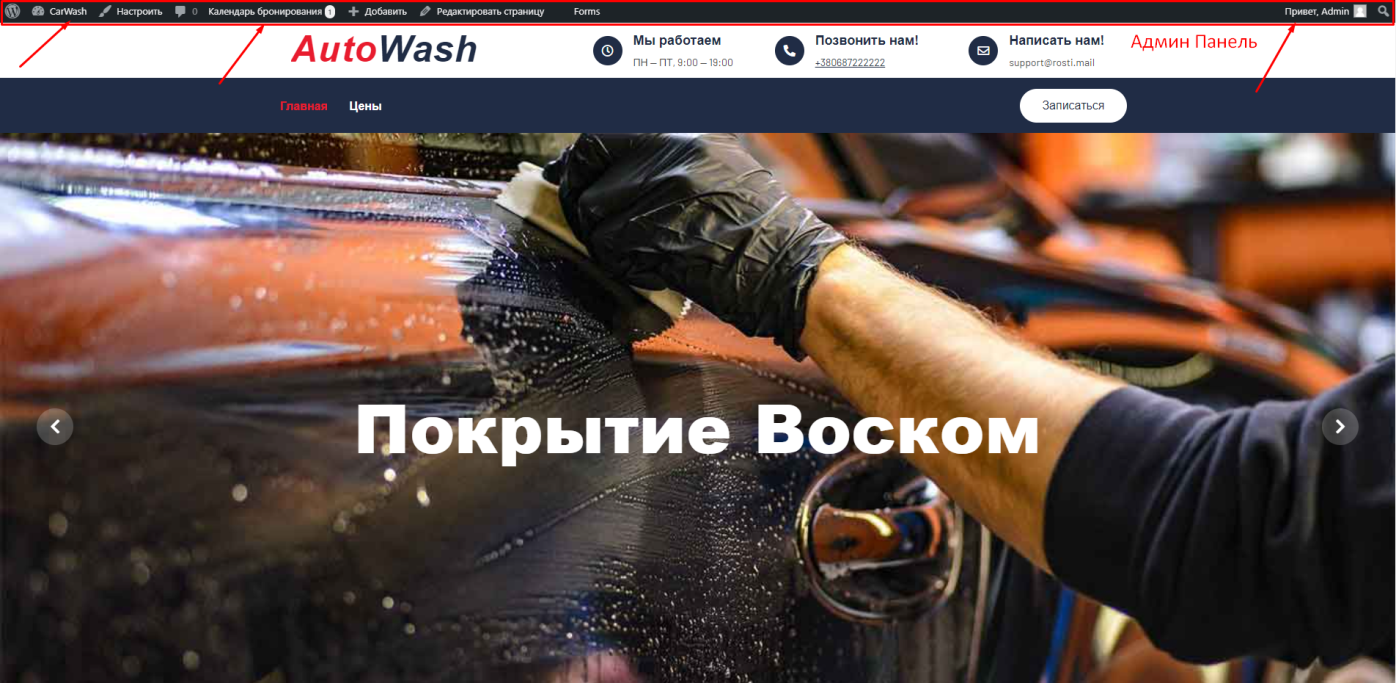


Рисунок 3.3 − Головна сторінка

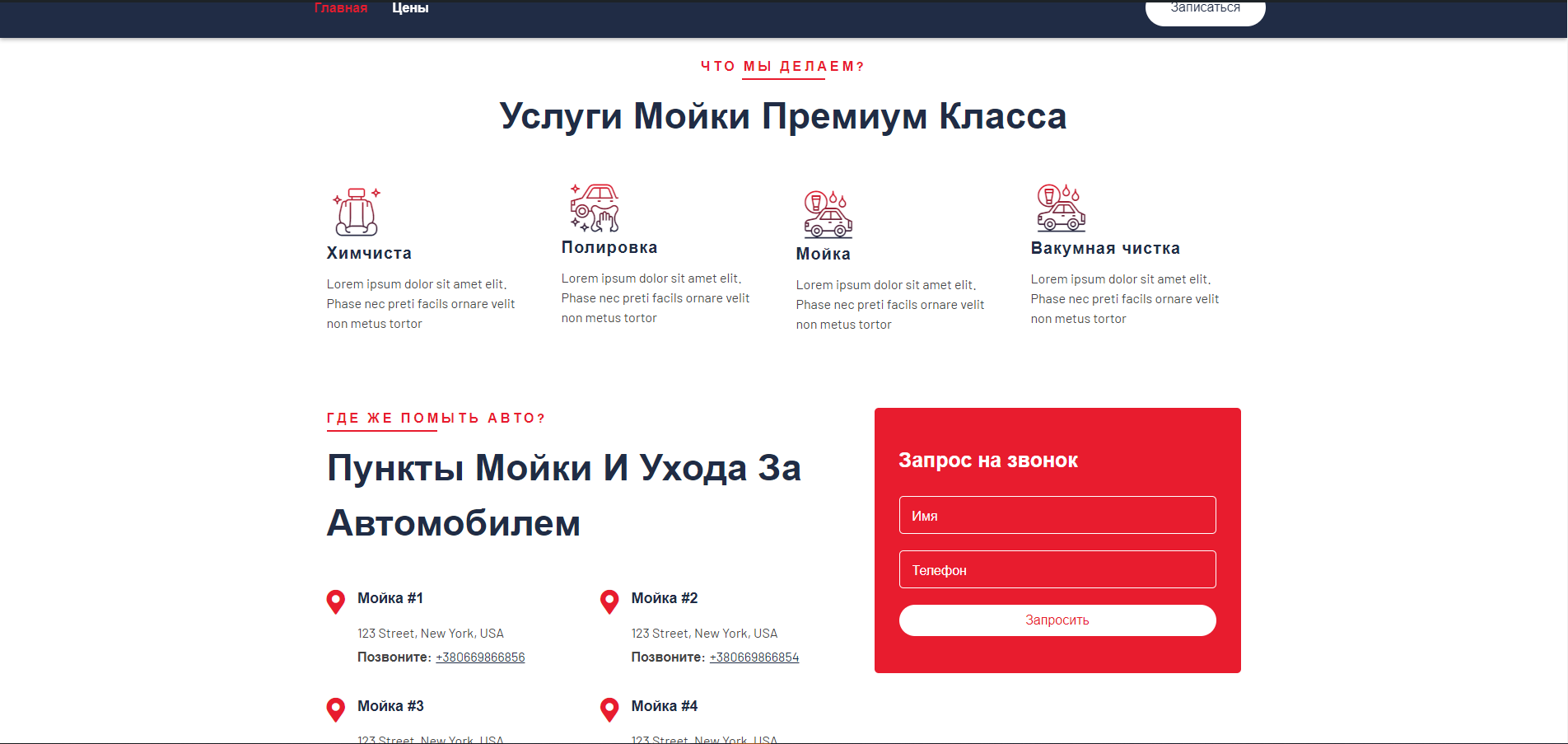


Рисунок 3.4 − Головна сторінка (продовження)

Головна сторінка містить наступні вкладки:

* Новости;
* Условия работ;
* О компании;
* Контакты.

Також головна сторінка містить класифікацію послуг для автомобілів, поділяються вони на 4 класи:

* Химчистка ;
* полировка;
* Мойка;
* Ваукумная очистка.

Сторінка автопарк (рис. 3.5)

На цій сторінці розміщені послуги для автомобілі, які можна детально переглянути (рис. 3.4), дізнатися цінові показники кожного з них, та замовити послуги, які вам найбільше підходять.

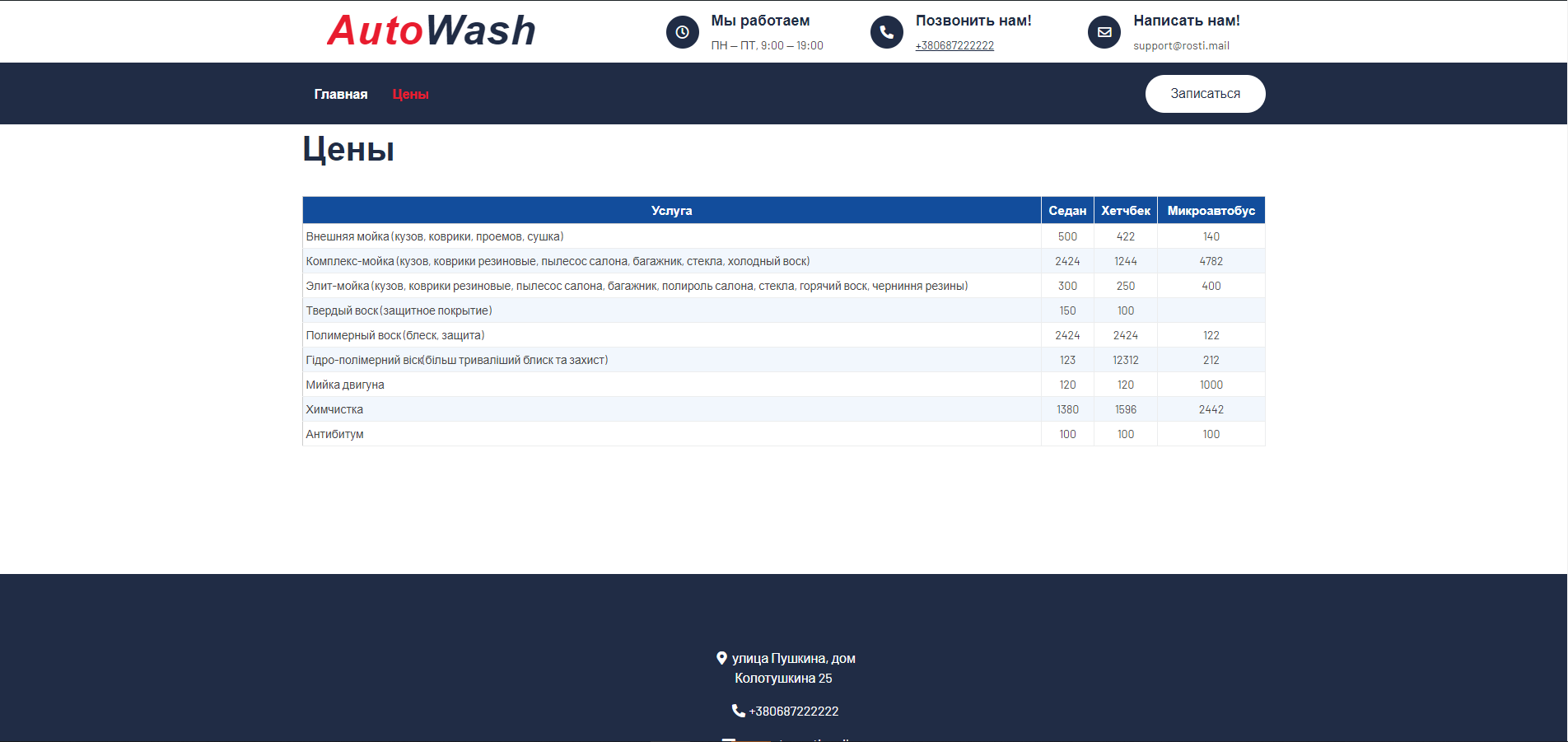


Рисунок 3.5 − Опис послуг

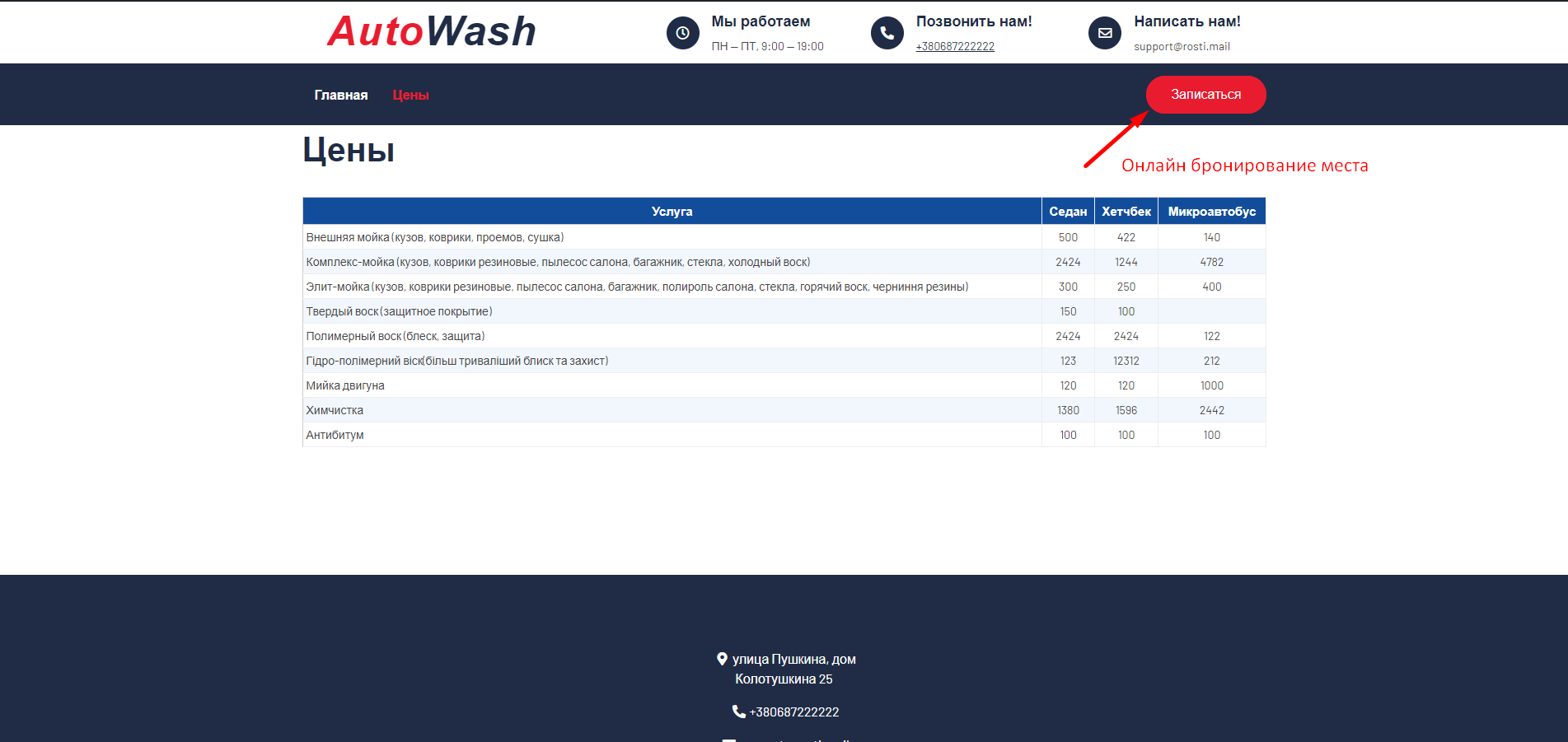


Рисунок 3.6 − Вид сторінки бронювання послуги

**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.3.7).

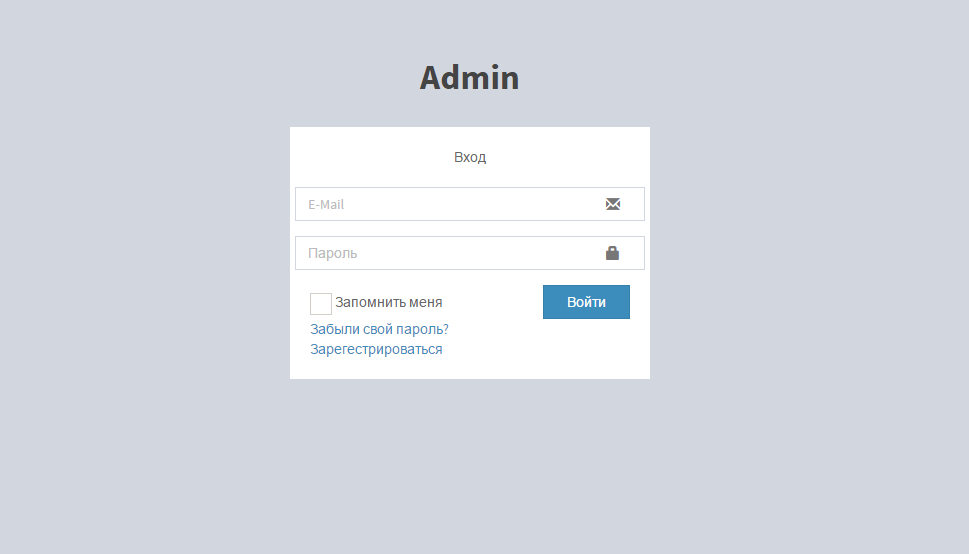
****

Рисунок 3.7 − 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. 3.8).

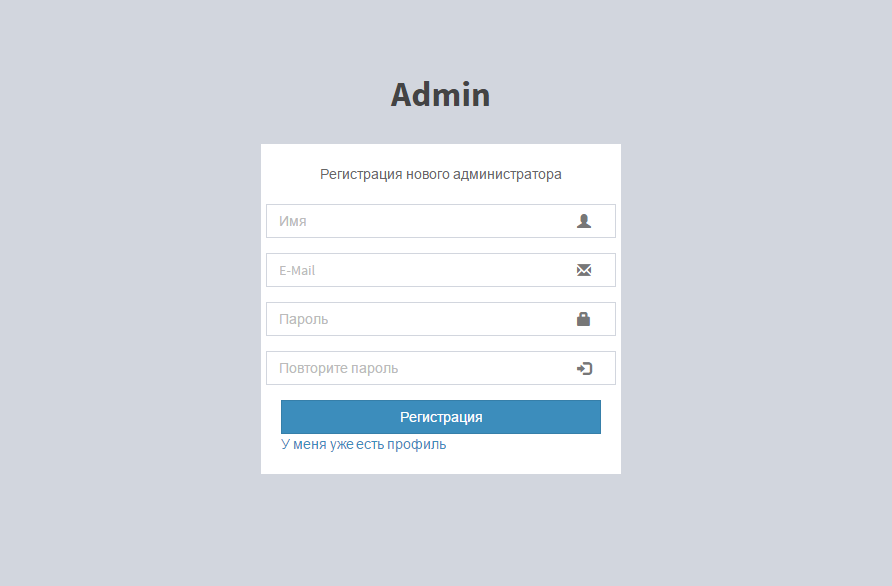


Рисунок 3.8 − Реєстрація нового адміністратора

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. 3.9). Бокове меню має 5 пунктів, а саме: На модерации, Принятые, Отклоненные, Добавление машин, Список машин.

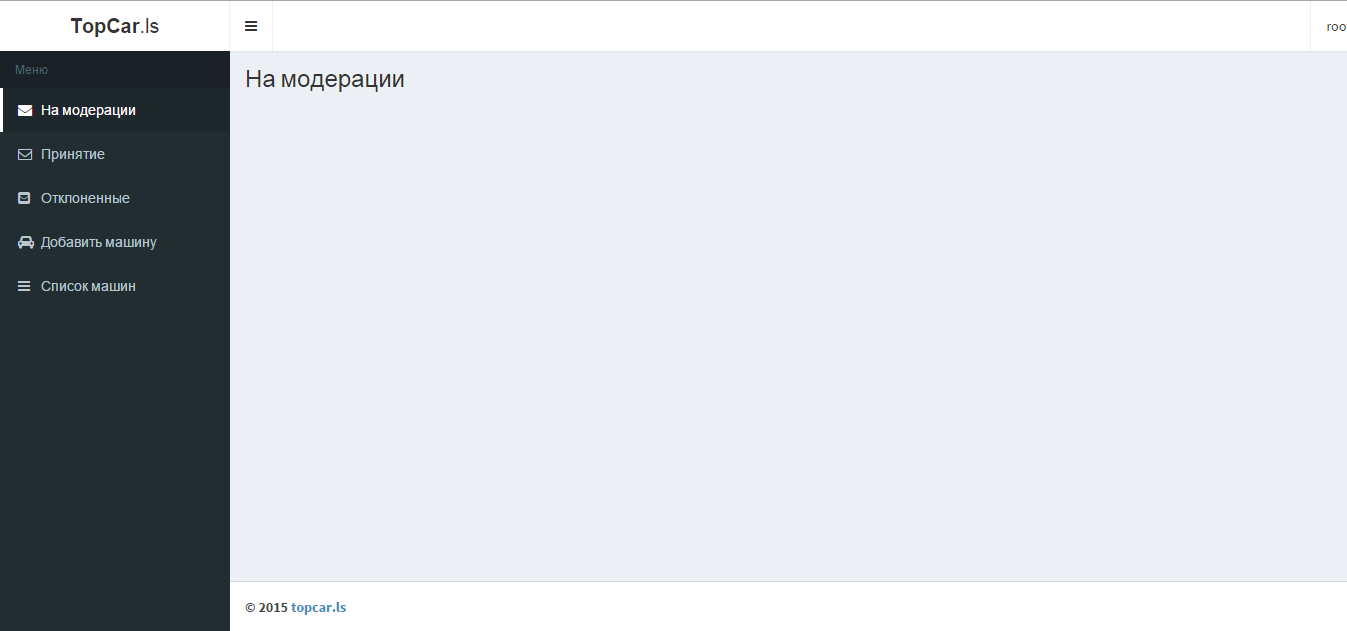


Рисунок 3.9 − Панель управління адміністратора

Зараз детально розглянемо кожен пункт меню бокової панелі. Почнемо з меню «На модерации». Перейшовши до цього меню адміністратор побачить повідомлення, які були надіслані користувачами. Після перегляду замовлення адміністратор повинен зважити та визначитися, надавати послугу замовнику чи все ж таки відмовити у наданні послуги з вагомих причин. В разі позитивного результату, адміністратор відправить повідомлення користувачеві на електронну пошту, про погодження надання послуги. В протилежному випадку, він принесе свої вибачення та вкаже на причини через які надання послуги неможливе. Приклад модерації (рис.3.10).

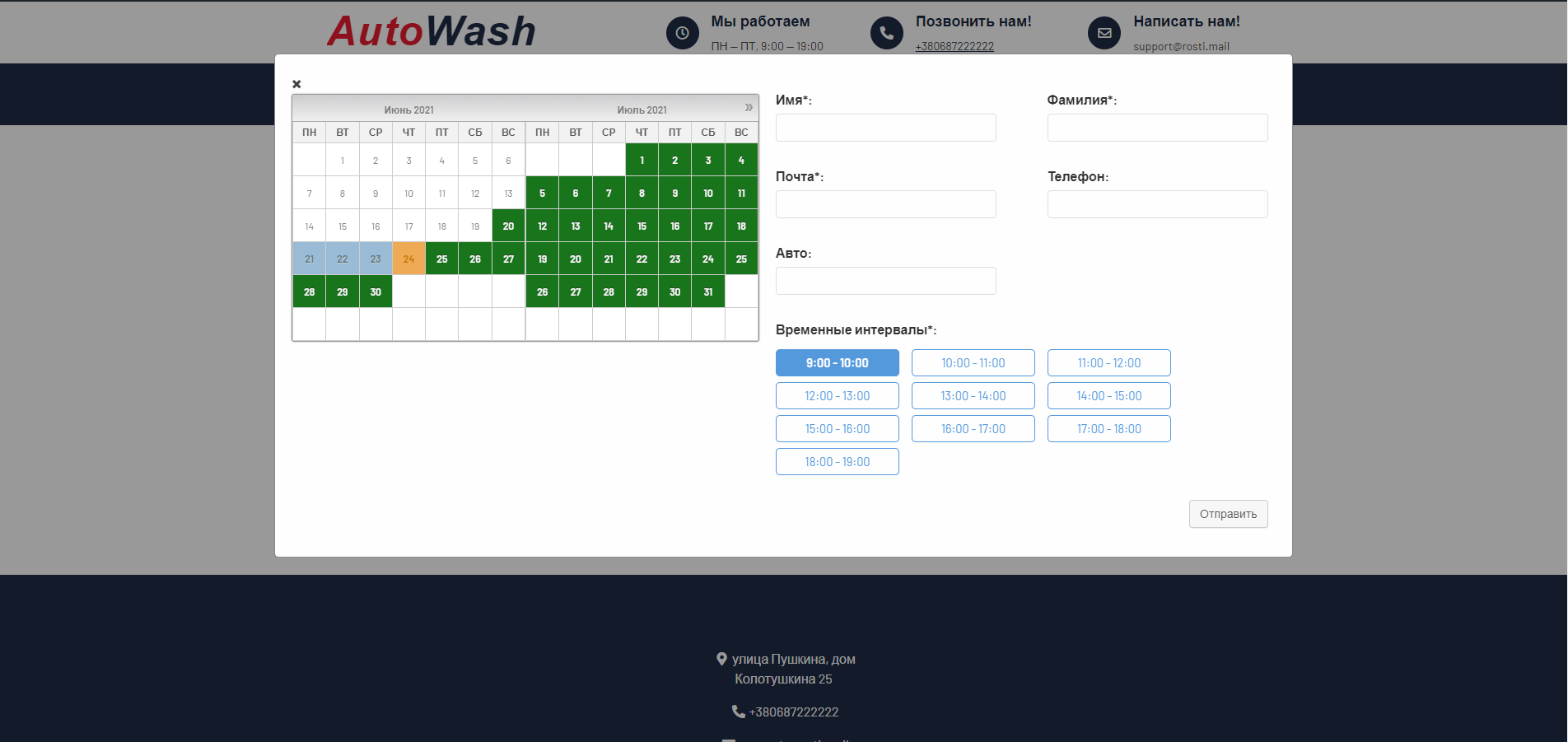


Рисунок 3.10 − Вид сторінки бронювання послуги в адмін. панелі

З наступними двома меню «Принятые» і «Отклоненные» здається все зрозуміло. Після перегляду та прийняття рішення адміністратором, повідомлення відправляються у відповідні розділи (рис. 3.11). Після того як адміністратор прийняв замовлення користувачеві було відправлено повідомлення про надання послуги.

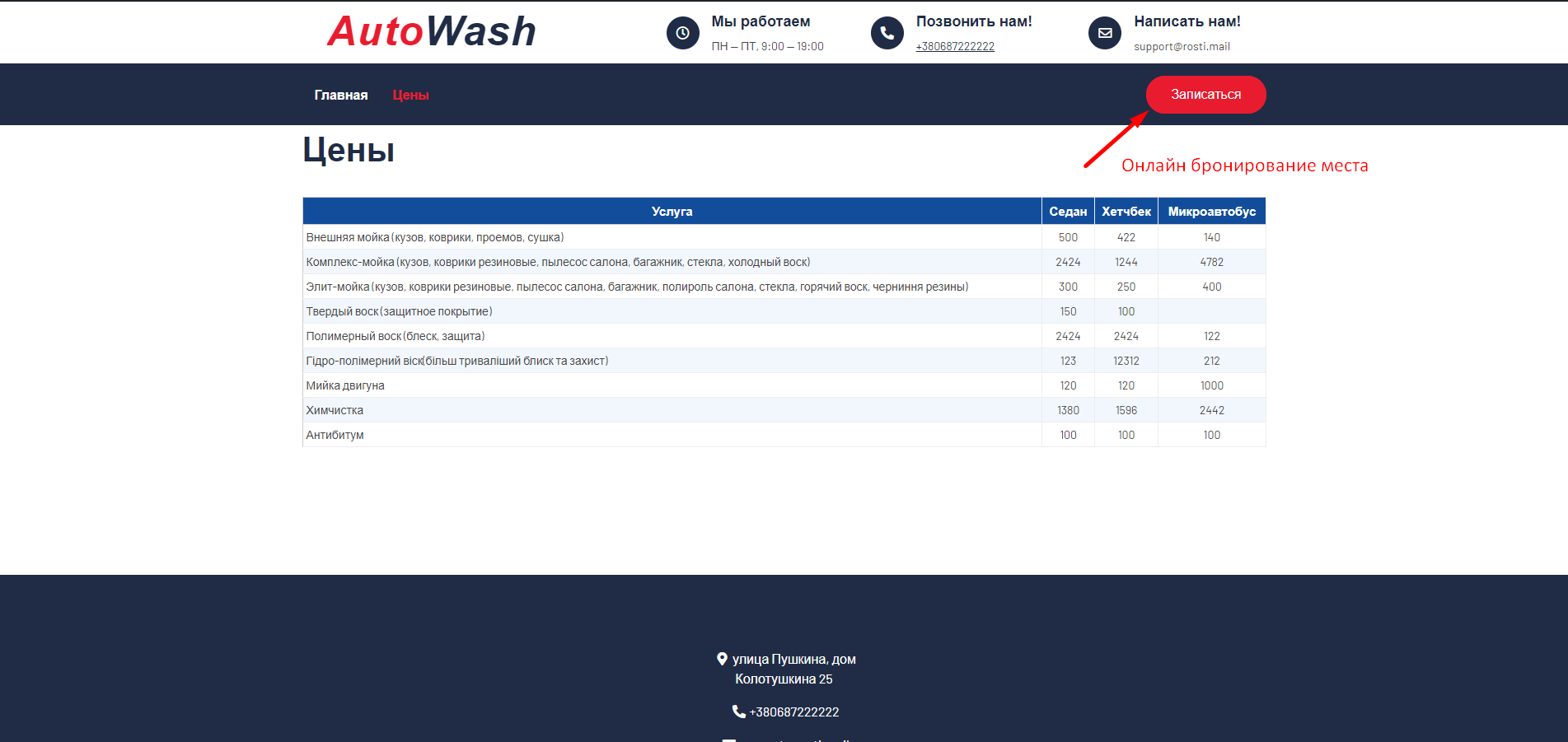


Рисунок 3.11 − вид форми бронювання

Приступимо до перегляду наступного розділу «Добавить машину». В цьому меню адміністратор може додати нове авто на сайт, заповнивши всі необхідні поля (рис.3.12).

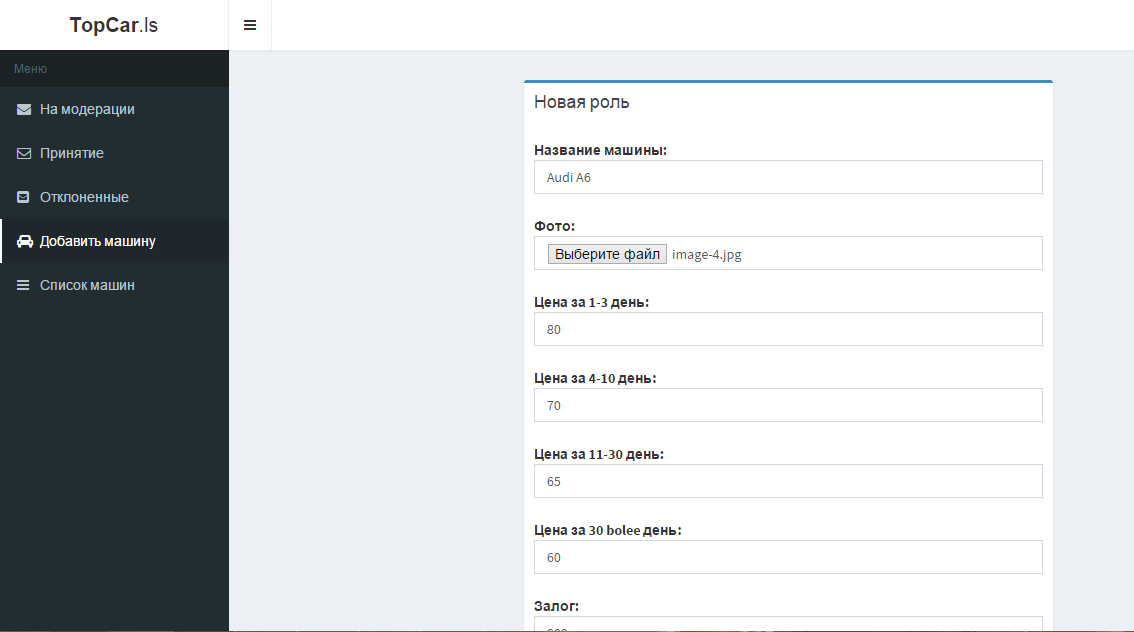


Рисунок 3.12 − Додавання нового автомобіля

Заповнивши всі необхідні поля, адміністратор повинен натиснути кнопку «Создать». Після всіх виконаних дій, послуги на автомобіль додані та готові до замовлення.

І останній пункт даного бокового меню «Список машин». В даному розділі знаходяться всі автомобілі, які були заброньовані адміністратором. Їх можна тимчасово видалити з сайту, редагувати, а також у потрібний момент відновити для перегляду на сайті (рис. 3.13).

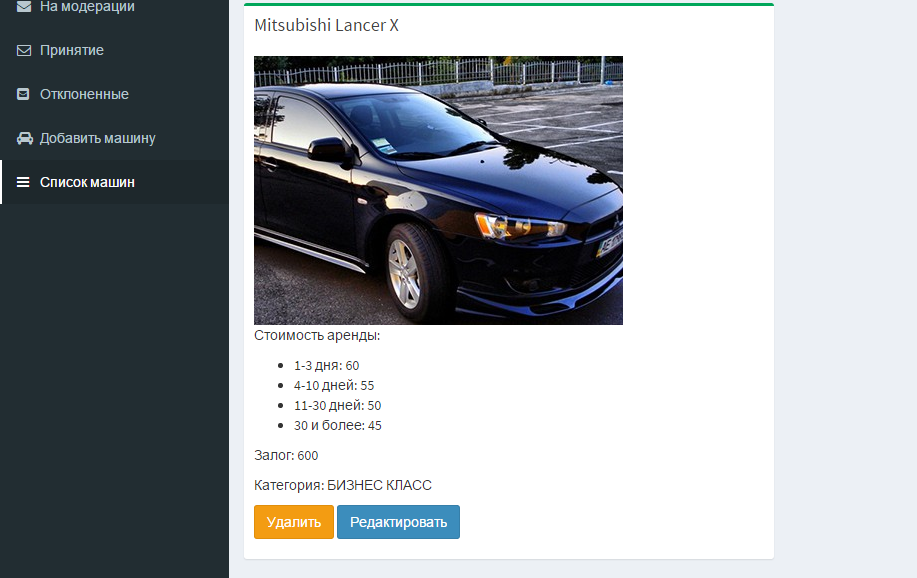


Рисунок 3.13 − Список автомобілів

**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.

З технічної точки зору тестування полягає у виконанні програми на деякій множині вихідних даних м звірці 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);
4. коли будете тестувати (послідовність проведення робіт: підготовка, тестування, аналіз результатів, в розрізі запланованих фаз розробки проекту);
5. критерії початку і закінчення тестування.

Тстування 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 маштабування сторінок | Успішно | Успішно | Успішно |
| Відображення в різних розширеннях | Успішно | Успішно | Успішно |
| Коректна зміна розміру текстових полів | Успішно | Успішно | Успішно |

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

Таблиця 4.2 – Тестування функціоналу

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

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

Таблиця 4.3 – Тест-кейси

|  |  |
| --- | --- |
| Тест-кейс №1. Тестування при невірному введенні логіну | |
| **Дії** | **Очікуваний результат** |
| 1. Відкриваємо інформаційну систему від імені адміністратора | 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 |

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

## ВИСНОВКИ

В даній дипломній роботі була спроектована та програмно реалізована інформаційна система з надання послуг для автомобілів м. Полтава. 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. Класифікація послуг для автомобілів на 4 класи спростить пошуки, а відсутність реєстрації значно зекономить час при визначенні та замовленні.

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

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гендина, Н.И. Использование формализованных методов при подготовке текста для веб-сайтов учреждений культуры [Текст] / Н.И. Гендина, Н.И. Колкова, О.И. Алдохина // Научные и технические библиотеки. – 2008. –№3. – С. 29-35
2. Иванов. А. Подготовка сайта для индексирования в поисковых системах [Електронний ресурс] / Андрей Иванов // Режим доступу: http://www.ashmanov.com/pap/ivsprep.phtm
3. Инькова, Н. А. Создание Web-сайтов: Учебно-методическое пособие [Електронний ресурс] / Инькова Н.А., Зайцева Е.А., Кузьмина Н.В, Толстых С.Г. // Режим доступу: <http://club-edu.tambov.ru/methodic/fio/p5.doc>
4. Комп’ютерні та Інформаційні технології і системи [Електронний ресурс] – режим доступу: http://fitts.pntu.edu.ua/ua/kafedry/inf-tekh-i-sy-m
5. Полтавський національний технічний університет імені Юрія Кондратюка [Електронний ресурс] – режим доступу: <http://pntu.edu.ua/>
6. Сайт харківського національного університету радіо-електроніки [Електронний ресурс] – режим доступу: <http://nure.ua/uk/>
7. Сайт київського політехнічного інституту імені Ігоря Сікорського [Електронний ресурс] – режим доступу: <http://kpi.ua/ru>
8. XHTML [Електронний ресурс] – режим доступу: [http://htmlbook.ru/ xhtml](http://htmlbook.ru/%20xhtml)
9. ANGULAR [Електронний ресурс] – режим доступу: <https://metanit.com/> web/angular/
10. APACHE [Електронний ресурс] – режим доступу: <http://www.apache.ru/>
11. MYSQL [Електронний ресурс] – режим доступу: <https://www.mysql.com/>
12. Діаграма прецедентів [Електронний ресурс] – режим доступу: <http://www.planerka.info/item/Diagramma-precedentov-(variantov-ispolzovaniya)-UML>
13. Діаграма діяльності [Електронний ресурс] – режим доступу: <http://it-gost.ru/articles/view_articles/96>
14. Контекстна діаграма [Електронний ресурс] – режим доступу: <http://techtrend.com.ua/index.php>
15. Аналіз баз даних [Електронний ресурс] – режим доступу: <https://www.osp.ru/os/2002/03/181272/>
16. Порівняльний аналіз СУБД [Електронний ресурс] – режим доступу: <https://sibac.info/studconf/tech/xxxvi/43413>
17. Порівняльний аналіз серверів [Електронний ресурс] – режим доступу: <http://www.it-ic.ru/sravnenie-veb-serverov-apache-iis>
18. Огляд web-серверів [Електронний ресурс] – режим доступу: <http://info-comp.ru/sisadminst/221-popular-web-servers.html>

21 Браун С. "Мозаїка" та "Всесвітня павутина" для доступу до Internet: Пер. c англ. – М.: Світ: Маліп: СК Пресс, 2008. – 167c.

22 Джефф Форрестол, Грег Шиплі, Сканери для виявлення вад у корпоративній мережі // Мережі і системи зв’язку – # 7. – 2009. – С.114 – 124.

23. Гришина Н. В. Організація комплексної системи захисту інформації. – М. Геліос АРВ, 2007. – 256с.

24. Сьомкін С. Н., Бєляков А. В., Грибанов С. В., Козачок В. І. Основи організаційного забезпечення інформаційної безпеки об’єктів інформатизації: Навчальний посібник – М .: Геліос АРВ, 2005. – 192с.

**ДОДАТОК А**

**КОД ПРОГРАМИ**

Код сторінки з цінами

<?php  
 /\*\*  
 \* Template Name: Price Page  
 \*/  
  
 get\_header(); ?>  
  
<div class="grid-container container">  
 <div class="grid-x">  
 <h1 class="page-title entry\_\_title"><?php the\_title(); ?></h1>  
  
 <table class="ceny">  
 <tr>  
 <th>Услуга</th>  
 <th>Седан</th>  
 <th>Хетчбек</th>  
 <th>Микроавтобус</th>  
 </tr>  
  
 <?php if ($prices = get\_field(‘prices’)): $i = 0 ?>  
 <?php foreach ($prices as $price): ?>  
 <tr>  
 <td><?php echo $price[‘name’] ?></td>  
 <td><?php echo $price[‘price’] ?></td>  
 <td><?php echo $price[‘price\_1’] ?></td>  
 <td><?php echo $price[‘price\_2’] ?></td>  
 </tr>  
 <?php endforeach; ?>  
 <?php endif; ?>  
  
 </table>  
 </div>  
</div>  
<?php get\_footer(); ?>

Код для попап-вікна бронювання місця

var scrollOut;  
 $( document ).ready(function() {  
  
// Get the modal  
 var modal = document.getElementById("myModal");  
  
// Get the button that opens the modal  
 var btn = document.getElementById("myBtn");  
  
// Get the <span> element that closes the modal  
 var span = document.getElementsByClassName("close")[0];  
  
// When the user clicks on the button, open the modal  
 btn.onclick = function() {  
 modal.style.display = "block";  
 }  
  
// When the user clicks on <span> (x), close the modal  
 span.onclick = function() {  
 modal.style.display = "none";  
 }  
  
// When the user clicks anywhere outside of the modal, close it  
 window.onclick = function(event) {  
 if (event.target == modal) {  
 modal.style.display = "none";  
 }  
 }

Код головної сторінки

<?php  
 /\*\*  
 \* Template Name: Home Page  
 \*/  
 get\_header(); ?>  
  
 <!--HOME PAGE SLIDER-->  
<?php home\_slider\_template(); ?>  
 <!--END of HOME PAGE SLIDER-->  
  
 <!-- BEGIN of main content -->  
 <div class="grid-container">  
 <div class="grid-x grid-margin-x">  
 <div class="cell">  
  
  
  
  
 <div class="service">  
 <div class="container">  
 <div class="section-header text-center">  
 <?php if ($title = get\_field(‘title’)): ?>  
 <p><?php echo $title; ?></p>  
 <?php endif; ?>  
 <?php if ($sub\_title = get\_field(‘subtitle’)): ?>  
 <h2><?php echo $sub\_title; ?></h2>  
 <?php endif; ?>  
 </div>  
 <?php if ($services = get\_field(‘services’)): $i = 0 ?>  
 <div class="row">  
 <?php foreach ($services as $service): ?>  
 <div class="col-lg-3 col-md-6">  
 <div class="service-item">  
 <?php $image = $service[‘serv\_icon’];  
 if ( ! empty($image)): ?>  
 <?php echo wp\_get\_attachment\_image($image[‘ID’], ‘full’, false, array(‘class’ => ‘icon’)); ?>  
 <?php endif; ?>  
 <h3><?php echo $service[‘name’] ?></h3>  
 <p><?php echo $service[‘info’] ?></p>  
 </div>  
 </div>  
 <?php endforeach; ?>  
 </div>  
 <?php endif; ?>  
 </div>  
 </div>  
  
 <div class="location">  
 <div class="container">  
 <div class="row">  
 <div class="col-lg-7">  
 <div class="section-header text-left">  
 <?php if ($title\_location = get\_field(‘title\_location’)): ?>  
 <p>  
 <?php echo $title\_location; ?>  
 </p>  
 <?php endif; ?>  
  
 <?php if ($subtitle\_location = get\_field(‘subtitle\_location’)): ?>  
 <h2>  
 <?php echo $subtitle\_location; ?>  
 </h2>  
 <?php endif; ?>  
 </div>  
  
 <?php if ($locations = get\_field(‘locations’)): $i = 0 ?>  
 <div class="row">  
 <?php foreach ($locations as $location): ?>  
 <div class="col-md-6">  
 <div class="location-item">  
 <i class="fa fa-map-marker-alt"></i>  
 <div class="location-text">  
 <h3><?php echo $location[‘name’] ?></h3>  
 <p><?php echo $location[‘adress’] ?></p>  
 <p>  
 <strong>Позвоните:</strong><a href="tel:<?php echo $location[‘phone’] ?>"><?php echo $location[‘phone’] ?></a>  
 </p>  
 </div>  
 </div>  
 </div>  
 <?php endforeach; ?>  
 </div>  
 <?php endif; ?>  
  
 </div>  
  
  
 <div class="col-lg-5">  
 <div class="location-form">  
 <h3>Запрос на звонок</h3>  
  
 <?php if (class\_exists(‘GFAPI’) && ($contact\_form = get\_field(‘call\_back’)) && is\_array($contact\_form)): ?>  
 <?php echo do\_shortcode("[gravityform id=‘{$contact\_form[‘id’]}’ title=‘false’ description=‘false’ ajax=‘true’]"); ?>  
 <?php endif; ?>  
  
 </div>  
 </div>  
 </div>  
 </div>  
 </div>  
  
 </div>  
  
 </div>  
 </div>  
 </div>  
 <!-- END of main content -->  
  
<?php get\_footer(); ?>

Код для хедера

<?php  
 /\*\*  
 \* Header  
 \*/  
?>  
<!DOCTYPE html>  
<html <?php language\_attributes(); ?>>  
<head>  
 <!-- Set up Meta -->  
 <meta http-equiv="X-UA-Compatible" content="IE=edge"/>  
 <meta charset="<?php bloginfo( ‘charset’ ); ?>">  
  
 <!-- Set the viewport width to device width for mobile -->  
 <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=5, user-scalable=yes">  
 <!-- Remove Microsoft Edge’s & Safari phone-email styling -->  
 <meta name="format-detection" content="telephone=no,email=no,url=no">  
  
 <!-- Add external fonts below (GoogleFonts / Typekit) -->  
 <!-- Google Font -->  
 <link href="https://fonts.googleapis.com/css2?family=Barlow:wght@400;500;600;700;800;900&display=swap" rel="stylesheet">  
  
 <!-- CSS Libraries -->  
 <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" rel="stylesheet">  
 <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css" rel="stylesheet">  
 <link href="lib/flaticon/font/flaticon.css" rel="stylesheet">  
 <link href="lib/animate/animate.min.css" rel="stylesheet">  
 <link href="lib/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">  
  
 <!-- Template Stylesheet -->  
 <link href="css/style.css" rel="stylesheet">  
  
 <?php wp\_head(); ?>  
  
</head>  
  
<body <?php body\_class( ‘no-outline’ ); ?>>  
<?php wp\_body\_open(); ?>  
  
<!-- <div class="preloader hide-for-medium">  
 <div class="preloader\_\_icon"></div>  
</div> -->  
  
<!-- BEGIN of header -->  
<div class="top-bar">  
 <div class="container">  
 <div class="row align-items-center">  
 <div class="col-lg-4 col-md-12">  
 <?php if ( $header = get\_field( ‘header’, ‘options’ ) ): ?>  
 <div class="logo">  
 <a href="<?php get\_permalink() ?>">  
 <h1><?php echo $header; ?></h1>  
 </a>  
 </div>  
 <?php endif; ?>  
  
 </div>  
 <div class="col-lg-8 col-md-7 d-none d-lg-block">  
  
 <?php if ( $short\_info = get\_field( ‘short\_info’, ‘options’ ) ): $i = 0 ?>  
 <div class="row">  
 <?php foreach ( $short\_info as $info ): ?>  
 <div class="col-4">  
 <div class="top-bar-item">  
 <div class="top-bar-icon">  
 <i class="<?php echo $info[‘icon’][‘value’] ?>"></i>  
 </div>  
 <div class="top-bar-text">  
 <h3><?php echo $info[‘title’] ?></h3>  
 <p><?php echo $info[‘info’] ?></p>  
 </div>  
 </div>  
 </div>  
 <?php endforeach; ?>  
 </div>  
 <?php endif; ?>  
  
 </div>  
 </div>  
 </div>  
</div>  
  
<header class="header nav-bar">  
 <div class="grid-container menu-grid-container">  
 <div class="grid-x grid-margin-x">  
 <div class="small-12 cell">  
 <?php if ( has\_nav\_menu( ‘header-menu’ ) ) : ?>  
 <div class="title-bar hide-for-medium" data-responsive-toggle="main-menu" data-hide-for="medium">  
 <button class="menu-icon" type="button" data-toggle aria-label="Menu" aria-controls="main-menu">  
 <span></span></button>  
 <div class="title-bar-title">Menu</div>  
 </div>  
 <nav class="top-bar" id="main-menu">  
 <?php wp\_nav\_menu( array(  
 ‘theme\_location’ => ‘header-menu’,  
 ‘menu\_class’ => ‘menu header-menu’,  
 ‘items\_wrap’ => ‘<ul id="%1$s" class="%2$s" data-responsive-menu="accordion medium-dropdown" data-submenu-toggle="true" data-multi-open="false" data-close-on-click-inside="false">%3$s</ul>‘,  
 ‘walker’ => new Foundation\_Navigation()  
 ) ); ?>  
 <button class="btn btn-custom btn-booking" type="button" id="myBtn">Записаться</button>  
 </nav>  
 <?php endif; ?>  
  
  
 <div id="myModal" class="modal">  
  
 <!-- Modal content -->  
 <div class="modal-content">  
 <span class="close">&times;</span>  
 <?php echo do\_shortcode(‘[booking type=1 nummonths=2 form\_type="standard"]’); ?>  
 </div>  
  
 </div>  
  
  
  
 </div>  
 </div>  
 </div>  
</header>  
<!-- END of header -->