

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інженерії програмного
забезпечення, канд.техн.наук, доцент,

_____Є.О. Давиденко

«___»_____2022 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

**ВЕБЗАСТОСУНОК МАГАЗИНУ АВТОМОБІЛЬНИХ
ТОВАРІВ**

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.01 – 408.21810805

Студент

_____Д. О. Безкровний

підпис

«___»_____2022 р.

Керівник д-р філ., викладач кафедри ІІЗ

_____К. О. Антіпова

підпис

«___»_____2022 р.

Консультант канд. техн. наук, доцент

_____А. О. Алексеєва

підпис

«___»_____2022 р.

Миколаїв – 2022

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ	3
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ІНФОРМАЦІЙНИХ ВЕБСАЙТІВ	5
1.1 Поняття інформаційного вебсайту	5
1.2 Використання вебсайтів для бізнесу	7
1.3 Огляд аналогів	9
1.4 Специфікація вимог до програмного забезпечення	12
Висновки до розділу 1	15
2 АНАЛІЗ ЗАСОБІВ РОЗРОБКИ	16
2.1 Етапи розробки вебзастосунків	16
2.2 Огляд програмного забезпечення.....	19
Висновки до розділу 2	24
3 АРХІТЕКТУРА, МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	25
3.1 Макет користувацького інтерфейсу	25
3.2 Інформаційна модель	28
3.3 Сценарії використання.....	29
Висновки до розділу 3	33
4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	34
4.1 Підготовка і створення HTML-сторінки.....	34
4.2 Наповнення сторінки та адаптивність.....	35
4.3 Зворотній зв'язок.....	36
4.3 Реєстрація та авторизація.....	38
4.4 Створення чат-бота	42
Висновки до розділу 4	44
ВИСНОВКИ	45
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	46

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ПЗ – програмне забезпечення

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

PHP – Hypertext Preprocessor

JS – JavaScript

AJAX – Asynchronous JavaScript And XML

ВСТУП

В останні роки можна спостерігати значні зміни в мережі інтернет. Сайти, які раніше були платформою для розташування статичного контенту, тепер стали багатофункціональними, інтерактивними системами для надання різної інформації.

На даний момент сайти є вже практично у всіх досить великих компаній та магазинів. Існують певні технології створення сайтів, якими потрібно добре володіти, щоб створити хороший, працює сайт.

Актуальність теми дослідження обумовлена збільшенням транспортних засобів у нашій країні та необхідністю аналізу існуючих методів розробки корпоративних і інформаційних ресурсів для виявлення найбільш ефективного методу їх створення.

Об'єктом роботи є процес створення Інтернет-сторінки для магазину.

Предметом роботи є програмні засоби реалізації вебзастосунку і його функцій.

Метою даної роботи є створення веб-застосунку для здобуття інформації про магазин і його послуги будь-яким користувачем, а також підвищення зручності зворотнього зв'язку. Для досягнення поставленої мети необхідно виконати наступні **завдання**:

- проаналізувати мови програмування, що будуть використовуватись для створення вебсайту;
- розробити концепцію майбутнього сайту;
- створити базу даних для інформаційної системи;
- реалізувати програмний код вебзастосунку;
- описати роботу з веб-застосунком.

Для реалізації поставлених завдань було використано наступні технології веб-розробки: мова програмування PHP, MySQL, HTML, CSS та JavaScript.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ІНФОРМАЦІЙНИХ ВЕБСАЙТІВ

1.1 Поняття інформаційного вебсайту

У сучасну цифрову епоху Інтернет вже переповнений різноманітними типами вебсайтів, і щодня у величезній кількості з'являються нові. Підраховано, що в Інтернеті понад 1,25 мільярда вебсайтів. Був час, коли сайти були лише текстовими і використовувалися виключно державними організаціями, сьогодні навіть найменші підприємства мають свій власний вебсайт.

Сайт, або web-сайт - сукупність логічно пов'язаних між собою веб-сторінок; також місце розташування контенту сервера. Зазвичай сайт в Інтернеті являє собою масив пов'язаних даних, що має унікальну адресу і сприймається користувачем як єдине ціле. Веб-сайти називаються так, тому що доступ до них відбувається по протоколу HTTP.

Вебсайти можна використовувати різними способами: особистий вебсайт, корпоративний вебсайт компанії, урядовий вебсайт, вебсайт організації тощо. Будь-який вебсайт може містити гіперпосилання на будь-який інший вебсайт, тому різниця між окремими сайтами, сприйнята користувачем, може бути розмитою.

Вебсайти можна розділити на дві великі категорії — статичні та інтерактивні. Інтерактивні сайти є частиною спільноти вебсайтів Web 2.0 і забезпечують інтерактивність між власником сайту та відвідувачами чи користувачами сайту. Статичні сайти подають або збирають інформацію, але не дозволяють безпосередньо взаємодіяти з аудиторією чи користувачами. Деякі вебсайти є інформаційними або створені ентузіастами або для особистого використання чи розваги. Багато вебсайтів прагнуть заробляти гроші, використовуючи одну або кілька бізнес-моделей, зокрема:

- розміщення цікавого контенту та продаж контекстної реклами;
- електронна комерція: продукти або послуги купуються безпосередньо через вебсайт;

- рекламні продукти чи послуги, доступні в звичайних підприємствах;
- базовий контент доступний безкоштовно, але преміум-контент вимагає оплати.

Статичний вебсайт — це вебсайт, веб-сторінки якого зберігаються на сервері у форматі, який надсилається клієнтському веб-браузеру. Аудіо чи відео може також вважатися "статичним" вмістом, якщо воно відтворюється автоматично або, як правило, неінтерактивне. Цей тип веб-сайтів зазвичай відображає однакову інформацію для всіх відвідувачів.. Незважаючи на те, що власник вебсайту може періодично оновлювати, це ручний процес редагування тексту, фотографій та іншого вмісту та може вимагати базових навичок дизайну вебсайту та програмного забезпечення.

Динамічний вебсайт – це вебсайт, який змінюється або налаштовується часто й автоматично. Динамічні сторінки на стороні сервера генеруються комп'ютерним кодом, який створює HTML.

Сайт може відображати поточний стан діалогу між користувачами, стежити за мінливою ситуацією або надавати інформацію, яка персоналізована відповідно до вимог окремого користувача. Динамічні сайти можуть бути інтерактивними, використовуючи форми HTML, зберігаючи та зчитуючи файли cookie браузера, або створюючи серію сторінок, які відображають попередню історію кліків.

Динамічний HTML використовує код JavaScript, щоб інструктувати веб-браузер, як інтерактивно змінювати вміст сторінки. Один із способів моделювання певного типу динамічного вебсайту, уникаючи втрати продуктивності при запуску динамічного механізму для кожного користувача або підключення — це періодично автоматично генерувати велику серію статичних сторінок.

Всі сайти в сукупності складають Всесвітню павутину, де комунікація (павутина) об'єднує сегменти інформації світової спільноти в єдине ціле – базу даних

і комунікації планетарного масштабу. Для прямого доступу клієнтів до сайтів на серверах був спеціально розроблений протокол НТТР.

1.2 Використання вебсайтів для бізнесу

Будь-який бізнес, який здійснюється завдяки взаємодії з широкою аудиторією, вимагає постійної реклами. Кілька років тому до звичайних засобів реклами додали новий, більш ефективний інструмент.

Починаючи з середини 90-х років, у всьому світі спостерігається зростання активності у сфері електронної торгівлі. Слідом за крупними компаніями-виробниками комп'ютерного обладнання в Інтернеті стали з'являтися торгівці традиційними товарами. Появилось велика кількість книжних магазинів, магазинів компакт-дисків і відеокасет, винних магазинів. Зараз практично будь-які товари можна купити через Інтернет.

Представляючи інформацію про себе та продуктах, фірма не тільки приваблює потенційних покупців, але й отримує зворотний зв'язок з ними. Крім цього, в Інтернеті простіше організувати адресну рекламу, коли веб-сайт рекламується серед людей, які можуть бути потенційними покупцями.

Довгий час символом представництва фірми була наявність веб-сайту, що вважалось показником її гнучкості та гарної здатності адаптуватися до змінюваних умов бізнесу.

В даний час багато фірм, розвивають свої сайти для покращення образу "обличчя" фірми у Всесвітній павутині. Вебсайти надають унікальну можливість для фірми ненав'язливо представляти великий обсяг інформації про себе, вироблених продуктів або наданих послуг. З іншої сторони, за інтересом до того чи іншого товару, зміни рейтингу наявного сайту можна судити про ефективність проведених рекламних акцій. Багато фірм, при випуску нових продуктів, спеціально створюють сайт з їх детальним описом, а потім через цей сайт отримують відгуки покупців і оперативно коригують свою діяльність.

Крім представленої опису товару, через Інтернет можна здійснити його продаж. Основна мета будь-якого комерційного сайту - обслуговування користувачів таким способом, який приносить компанію найбільший прибуток.

Доцільність створення того чи іншого типу сайту для бізнесу визначають круг вирішуваних сайтом задач, де необхідно вибрати основну мету створення сайту та другорядні цілі та бюджет сайту.

Власники малого бізнесу зазвичай мають невеликий бюджет, оскільки їх прибуток буде зростати лише з часом. Однак їм необхідно налагодити ефективну присутність в Інтернеті вже зараз, на початковому етапі свого бізнесу, щоб мати можливість отримувати пристойний прибуток у майбутньому. Існує причинно-наслідковий зв'язок.

Незалежно від початкового капіталу, низькобюджетна веброзробка дозволяє отримати простий, але ефективний вебсайт або застосунок, який може перетворювати відвідувачів у клієнтів. Для стартапів і малого бізнесу цілком доступно створити мінімально життєздатний продукт, щоб швидко з'явитися на ринку, а потім інвестувати в його покращення.

Окрім охоплення бізнесу та залучення клієнтів, професійний вебсайт також надає багато інших цінних переваг будь-якому бізнесу чи компанії. Деякі з цих переваг включають підвищення довіри до бізнесу, збільшення обсягу продажів за рахунок залучення більшої кількості клієнтів, надання можливості бізнесу вийти на міжнародний ринок, покращене та ефективне обслуговування клієнтів через онлайн-форуми та зниження операційних витрат ділової активності.

Основною причиною розробки зручних вебсайтів є простота взаємодії між вебсайтом та його користувачами. Якщо компанія створює дуже важкий або складний вебсайт, це не тільки втратить дорогоцінний час користувачів, але й призведе до зниження продуктивності для компанії. Клієнти є основним джерелом продуктивності для будь-якої компанії, і якщо клієнти не залучаються до компанії,

вона ніколи не зможе досягти своїх бажаних цілей і завдань. Тому дуже важливо, щоб компанія була орієнтована на клієнта.

Якщо компанія займається онлайн-бізнесом, вона повинна створити такий веб-сайт для маркетингу продуктів, який зможе залучати клієнтів. А досягти цієї мети можна лише створивши зручний для користувача вебсайт. Існує багато факторів і питань, які слід враховувати при плануванні створення зручного вебсайту. Ці фактори включають зовнішній вигляд вебсайту, навігацію та елементи керування, доступ до сторінок, макет і дизайн, а також швидкість завантаження. Якщо під час розробки веб-сайту залишити будь-який з цих факторів, основна мета створення зручного для користувача веб-сайту ніколи не буде досягнута.

1.3 Огляд аналогів

Для аналізу сучасного стану предметної області проведено дослідження вебзастосунків магазинів автотоварів. Для аналізу було обрано декілька популярних компаній, таких як:

1. Fastavto

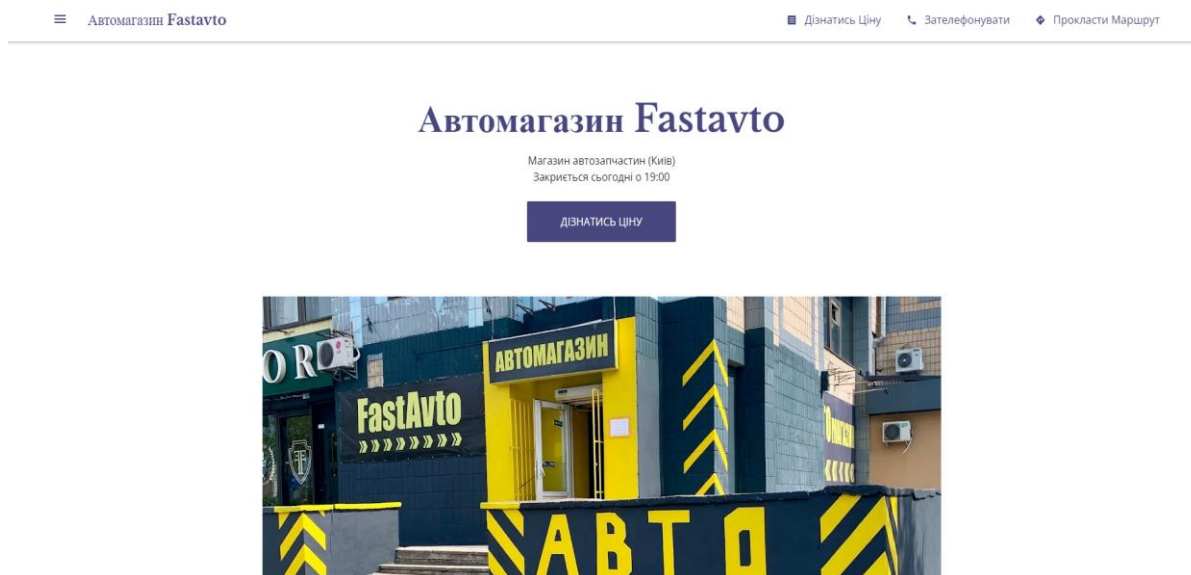


Рисунок 1.1 – інтерфейс сайту Fastavto

Основні характеристики існуючого ПЗ:

- домен: fastavto.business.site;
- архітектура: client-server;
- мова реалізації: PHP, JavaScript
- перелік функцій, характеристик: пошук основної інформації про магазин, онлайн-запис, демонстрація відгуків.
- аналіз переваг та недоліків даного ПЗ:
 - переваги: простий інтерфейс, швидкість загрузки.
 - недоліки: невелика кількість функцій .

2. Exist.ua

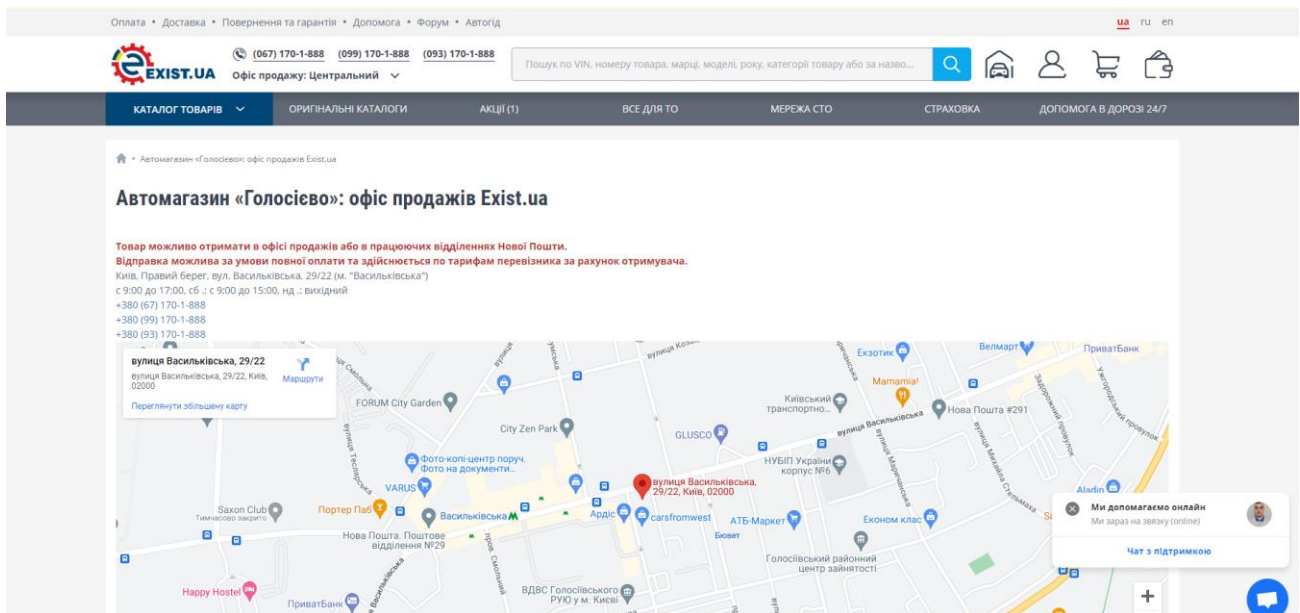


Рисунок 1.2 – інтерфейс сайту Exist.ua

Основні характеристики існуючого ПЗ:

- домен: exist.ua;
- архітектура: client-server;
- мова реалізації: PHP, JavaScript

- перелік функцій, характеристик: пошук основної інформації про магазин, онлайн-оператор, доступ до каталогу товарів, можливість реєстрації;
- аналіз переваг та недоліків даного ПЗ:
 - переваги: простий інтерфейс, швидкість загрузки.
 - недоліки: невелика кількість функцій, складність пошуку необхідного товару.

3. ATL

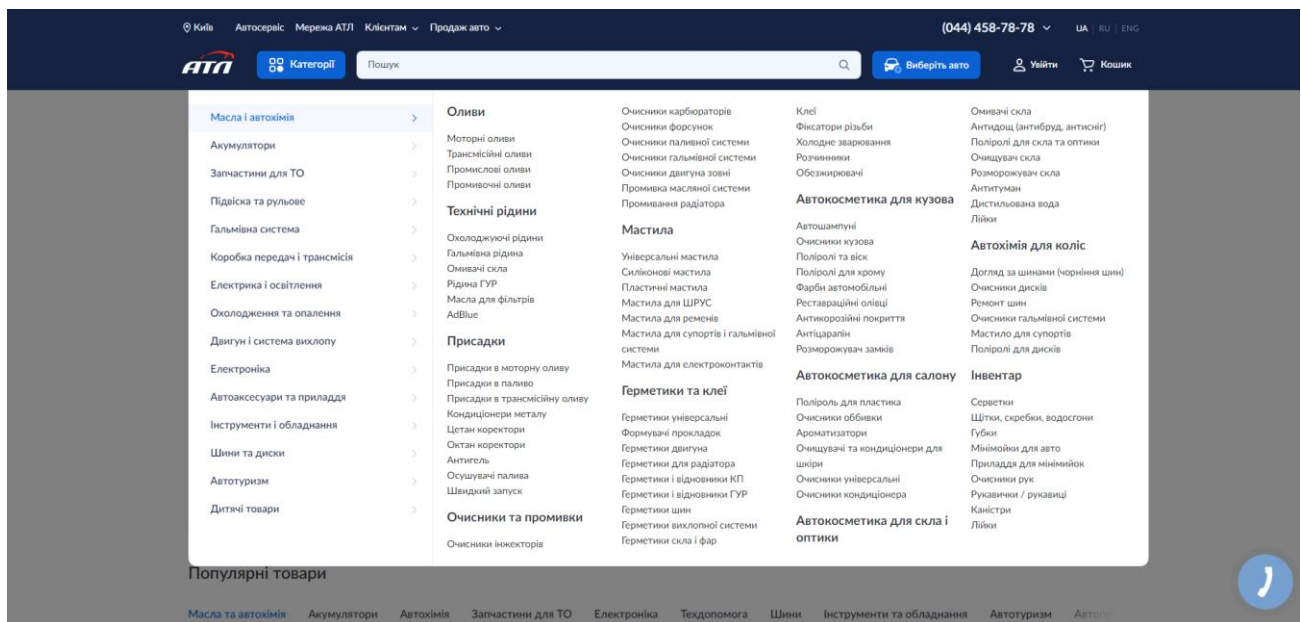


Рисунок 1.3 – інтерфейс сайту ATL

Основні характеристики існуючого ПЗ:

- домен: atl.ua;
- архітектура: client-server;
- мова реалізації: Java, Python, Ruby, Node.Js;
- перелік функцій, характеристик: замовлення товарів, онлайн-оператор, доступ до каталогу товарів, можливість реєстрації, система сповіщення;
- аналіз переваг та недоліків даного ПЗ:
 - переваги: велика кількість функцій, можливість замовлення товару;

- недоліки: незручний інтерфейс, повільна загрузка.

1.4 Специфікація вимог до програмного забезпечення

ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ

1. Призначення системи (застосунку), для якої розробляється програмне забезпечення

Призначенням застосунку є удосконалення зворотнього зв'язку та підвищення зручності у здобуванні інформації про товари та послуги офлайн-магазину автотоварів.

2. Межі проєкту ПЗ

Завершення роботи над програмним забезпеченням заплановано на 15.06.2022

ЗАГАЛЬНИЙ ОПИС

1. Сфера застосування

Дане програмне забезпечення може бути використаним підприємцями, яких не влаштовує їх поточний вебсайт, або його не мають.

2. Характеристики користувачів

Основні характеристики користувачів: наявність девайсу підключеного до мережі Інтернет (ПК, планшет, смартфон).

3. Загальні обмеження

Стабільний доступ до мережі Інтернет.

ФУНКЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Функція реєстрації/ авторизації

Опис функції

Функція дає можливість користувачу зареєструватись у системі та виводити вхід у майбутньому.

Вхідна і вихідна інформація

Вхідна інформація – дані користувача для реєстрації.

Вихідна інформація – доступ до вебсайту.

Функціональні вимоги

Доступ до мережі Інтернет та бази даних системи.

Функція зворотнього зв'язку

Опис функції

Функція призначена для відправки електронних листів до власника сайту.

Вхідна і вихідна інформація

Вхідна інформація – дані користувача, текст повідомлення, електронна адреса.

Вихідна інформація – відправлений лист.

Функціональні вимоги

Доступ до мережі Інтернет.

Функція чат-бота

Опис функції

Інтеграція чат-бота до сайту дозволить користувачам дізнатися необхідну інформацію без пошуку та втрачання часу.

Вхідна і вихідна інформація

Вхідна інформація – одне з питань на вибір користувача;

Вихідна інформація – відповідь на запитання користувача.

Функціональні вимоги

База даних чат-бота та доступ до мережі Інтернет.

ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

1. Джерела і зміст вхідної інформації (даних)

Основним джерелом вхідної інформації є сам користувач, а також база даних системи.

ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Основною вимогою до програмного забезпечення є підтримка актуальних версій сучасних браузерів та стабільних доступ до мережі Інтернет.

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1. Системне програмне забезпечення

Основні функції вебсайту мають бути створені за допомогою JavaScript, PHP, HTML, CSS. Для роботи з базами даних має бути використаний MySQL.

2. Мережне програмне забезпечення

Для створення ПЗ використано ОС Windows 10, середовищем розробки було використано застосунок Sublime Text, для зручного перегляду вебсторінок – Google Chrome.

3. Мова і технологія розробки ПЗ

У якості мови розробки було використано JavaScript PHP.

ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ

1. Інтерфейс користувача

Інтерфейс повинен бути інтуїтивно зрозумілим для звичайного користувача

2. Апаратний інтерфейс

Апаратним інтерфейсом є девайс користувача (ПК, смартфон чи планшет)

ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1. Доступність

Вебзастосунок є цілком доступним для кожного користувача, який має девайс з доступом до мережі Інтернет.

2. Переносимість

Вебзастосунок повинен працювати на усіх девайсах, які задовольняють мінімальні вимоги для підтримки браузера.

3. Продуктивність

Продуктивність програмного забезпечення залежить від швидкості підключення до мережі Інтернет.

4. Надійність

Особисті дані користувачів повинні бути приховані від інших користувачів.

5. Безпека

Безпека вебсайту залежить від умов хостингу, на якому розташований вебсайт.

Висновки до розділу 1

У розділі 1 було проведено аналіз предметної області. Було з'ясовано значення поняття «вебсайт» та виокремлено його різновиди.

Також було проаналізовано наявні аналоги. На основі проведеного аналізу було з'ясовано, що інформаційні сторінки магазині широко використовуються. Кожен розглянутий аналог має свої переваги та недоліки. Актуальність розробки власного інформаційного порталу полягає також у викоріненні таких мінусів: повільна загрузка сторінки, складний інтерфейс, відсутність онлайн-помічника.

На основі першого розділу можна приступати до проєктування вебсторінки магазину автотоварів.

2 АНАЛІЗ ЗАСОБІВ РОЗРОБКИ

2.1 Етапи розробки вебзастосунків

1. Збір інформації: мета, основні цілі та цільова аудиторія

Цей етап, етап виявлення та дослідження, визначає, як виглядатимуть наступні етапи. Найважливішим завданням на цьому етапі є чітке розуміння цілей майбутнього вебсайту, та цільової аудиторії, яку потрібно залучити на сайт.

Портал новин відрізняється від розважальних сайтів, а онлайн-ресурси для підлітків виглядають інакше, ніж сайти для дорослих. Різні типи вебсайтів надають відвідувачам різну функціональність, а це означає, що різні технології повинні використовуватися відповідно до цілей. Добре описаний і детальний план, заснований на цих даних перед розробкою, може захистити від витрачання додаткових ресурсів на вирішення несподіваних проблем, таких як зміна дизайну або додавання функціональних можливостей, які спочатку не планувалися.

2. Планування: створення карти сайту.

На цьому етапі розробки вебсайту розробник створює дані, які дозволяють замовнику судити, як буде виглядати весь сайт. На основі інформації, яка була зібрана на попередньому етапі, створюється карта сайту.

Карта сайту повинна описувати зв'язки між основними областями веб-сайту. Таке уявлення може допомогти зрозуміти, наскільки придатним буде кінцевий продукт. Він може показати «зв'язок» між різними сторінками веб-сайту, щоб можна було судити, наскільки легко кінцевому користувачеві буде знайти потрібну інформацію чи послугу, якщо він почне з головної сторінки. Основною причиною створення карти сайту є створення зручного та легкого в навігації вебсайту.

Карта сайту дозволяє зрозуміти, як виглядає внутрішня структура вебсайту, але не описує інтерфейс користувача.

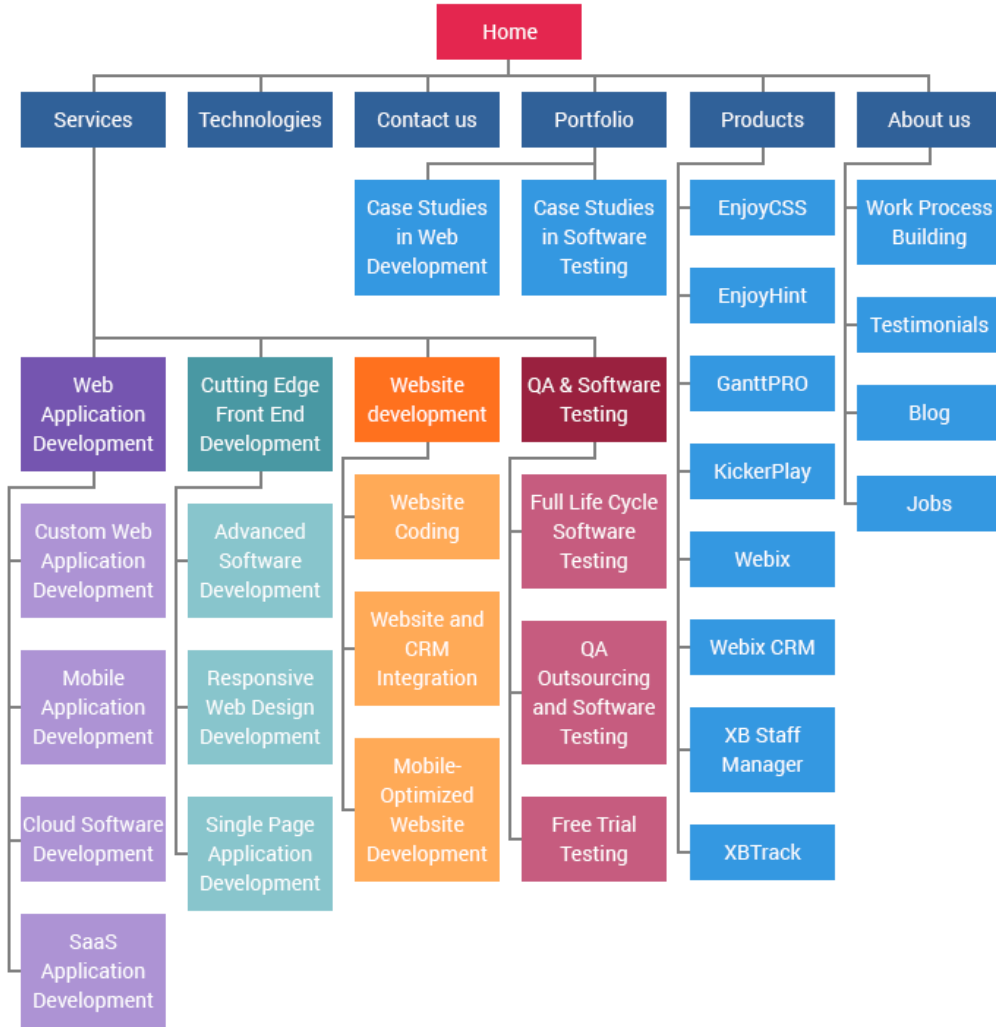


Рисунок 1.4 – карта інформаційного вебсайту

3. Дизайн: макети сторінки, огляд і цикл затвердження

На етапі дизайну ваш веб-сайт набуває форму. На цьому кроці створюється весь візуальний вміст, наприклад зображення, фотографії та відео. Вся інформація, яка була зібрана на першому етапі, має вирішальне значення. Під час роботи над дизайном необхідно пам'ятати про клієнтів і цільову аудиторію.

Макет сайту – це результат роботи дизайнера. Це може бути графічний ескіз або реальний графічний дизайн. Основною функцією макета є представлення інформаційної структури, візуалізації вмісту та демонстрації основної

функціональності. Макети містять кольори, логотипи, зображення і можуть дати загальне уявлення про майбутній продукт.



Рисунок 1.5 – макети сторінок вебсайту

4. Написання та збір контенту

Написання та компіляція контенту зазвичай перетинаються з іншими етапами створення вебсайту, і його роль не можна недооцінювати. На цьому кроці необхідно письмово записати суть, яку треба донести до аудиторії вебсайту, і додати заклики до дії. Написання контенту також передбачає створення заголовків, редагування тексту, написання нового тексту, компіляцію існуючого тексту тощо, що вимагає часу та зусиль.

5. Кодування

На цьому кроці можна приступити до створення самого вебсайту. Графічні елементи, які були розроблені на попередніх етапах, слід використовувати для

створення фактичного вебсайту. Зазвичай спочатку створюється домашня сторінка, а потім додаються всі підсторінки відповідно до ієрархії вебсайтів, яка раніше була створена у вигляді карти сайту. Фреймворки та CMS повинні бути впроваджені, щоб переконатися, що сервер може безперебійно виконувати встановлення та налаштування.

Усі статичні елементи вебсторінки, які були розроблені під час створення макета, повинні бути створені та протестовані. Потім слід додати спеціальні функції та інтерактивність.

6. Тестування, огляд і запуск

На цьому етапі слід перевірити кожне посилання, кожну форму, кожен сценарій, запустити програмне забезпечення для перевірки орфографії, щоб знайти можливі помилки.

Після перевірки треба завантажити вебсайт на сервер. Для цього використовується програмне забезпечення FTP (File Transfer Protocol).

7. Технічне обслуговування: моніторинг думки та регулярне оновлення

Важливо пам'ятати, що веб-сайт – це послуга, а не продукт. Крім оприлюднення вебсайту, треба постійно перевіряти, що все працює як слід.

Додана на сайт система зворотного зв'язку дозволить виявити можливі проблеми, з якими стикаються кінцеві користувачі. Найвищим пріоритетом у цьому випадку є вирішення проблеми якомога швидше. Якщо ви цього не зробите, одного разу ви можете виявити, що ваші користувачі вважають за краще використовувати інший веб-сайт, а не миритися з незручностями.

2.2 Огляд програмного забезпечення

PHP-скрипт

Це скриптова мова програмування, створена для генерації HTML-сторінок на веб-сервері і роботи з базами даних. На даний момент він підтримується практично

всіма представниками хостингу, входить в «стандартний» набір для створення сайтів (LAMP - Linux, Apache, MySQL, PHP).

Завдяки своїй простоті, швидкості виконання, багатій функціональності, розповсюдженню початкових кодів на основі ліцензії PHP, ця мова є чи не найпопулярнішим в області технологій створення сайтів. Відрізняється наявністю ядра і модулів, «розширень»: для роботи з базами даних, сокетами, динамічною графікою, криптографічними бібліотеками, документами формату PDF і т.п. Є можливість розробити, а також підключити додаткове розширення.

Можливості PHP дуже великі. Головним чином, PHP застосовується при написанні скриптів, що працюють на стороні сервера; таким чином, PHP здатний виконувати все те, що виконує будь-яка інша програма CGI (наприклад, обробляти дані форм, генерувати динамічні сторінки, відсилати і приймати cookies). Але PHP дає можливість виконувати також безліч інших завдань.

Існують три основні області, де використовується PHP:

- створення скриптів для виконання на стороні сервера;
- створення скриптів для виконання в командному рядку;
- створення додатків GUI, що виконуються на стороні клієнта.

Крім цього, PHP:

- доступний для більшості операційних систем, включаючи Linux, багато модифікації Unix (такі, як HP-UX, Solaris і OpenBSD), Microsoft Windows, Mac OS X, RISC OS, і багатьох інших;
- включає підтримку більшої частини веб-серверів (для більшості серверів PHP поставляється в якості модуля, для інших, що підтримують стандарт CGI, PHP може функціонувати в якості процесора CGI);
- підтримує широке коло баз даних;

- підтримує DBX для роботи на абстрактному рівні (таким чином можна працювати з будь-якою базою даних, що використовують DBX); ODBC (тобто ви можете працювати з будь-якою базою даних, що підтримує цей стандарт);
- підтримує "спілкування" з іншими сервісами з використанням різних протоколів: LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (на платформах Windows), а також багатьох інших;
- підтримує стандарт обміну складними структурами даних WDDX;
- підтримує об'єкти Java, дає можливість використовувати їх в якості об'єктів PHP;
- дає можливість формувати зображення, файли PDF, ролики Flash, створені "на льоту"; здатний видавати будь-які текстові дані (XHTML, інші XML-файли); автоматично генерувати і зберігати в файлової системі вашого сервера;
- включає засоби обробки текстової інформації, починаючи з регулярних виразів Perl або POSIX Extended і закінчуючи парсером документів XML;
- підтримує багато інших розширення (функції пошукової машини mnoGoSearch, функції IRC Gateway, функції для роботи зі стиснутими файлами (gzip, bz2), функції календарних обчислень, функції перекладу та ін.).

JavaScript

Це поки ще відносно молодий мову програмування, але вже дуже популярний в області технологій створення сайтів. На даний момент, робота над ним ще не закінчена. Він постійно допрацьовується і удосконалюється. Технічний комітет працює над істотними розширеннями, включаючи механізми для сценаріїв, які будуть створені для застосування в Internet, а також більш жорсткою координацією з іншими основними стандартами груп World Wide Web Консорціум і Wireless Application Protocol Форум. Java Script вже зіграв дуже важливу роль у розвитку технологій створення сайтів

В даний час використовується третє видання ECMA-262, що включає потужні регулярні вирази, кращу обробку рядків, нові інструкції контролю, управління, перехоплення і обробку виняткових ситуацій, більш жорстке визначення помилок, форматування для числового виведення і незначні зміни в очікуванні введення засобів багатомовності та майбутнього розвитку мови.

HTML

Ця мова є базовим в області технологій створення сайтів, так як відносно легкий в освоєнні. Але надмірна простота є і його недоліком. HTML (від англійського Hyper Text Markup Language - мова розмітки гіпертексту) прекрасно відповідав вимогам раннього періоду розвитку технологій створення сайтів, але з подальшим його розвитком виникли істотні проблеми. HTML надає наступні можливості:

- видавати мережеві документи з заголовками, текстом, таблицями, списками, фотографіями і т.п. ;
- отримувати інформацію з Мережі через посилання гіпертексту при натисканні кнопки;
- створювати форми для посилки запитів на віддалені комп'ютери, щоб проводити пошук інформації, здійснювати бронювання, замовляти товари та т.п. ;
- включати електронні таблиці, відео кліпи, аудіо кліпи, і інші програмні додатки безпосередньо в їх документи.

Історія розробки HTML досить тривала. У кожній його версії розробники намагалися домогтися того, щоб HTML-сторінки читалися усіма браузерами, на всіх комп'ютерних платформах.

Мова HTML призначений для формування та оформлення сторінок тексту, який призначений для перегляду в мережі Інтернет за допомогою спеціальних програм - браузерів (основні з них Netscape Navigator (фірми Netscape) і Internet Explorer (фірми Microsoft)).

В даному випадку, під сторінкою мається на увазі окремий файл зі спеціальними позначками, які вказують браузеру, як саме повинен бути відображений на екрані і в роздруківці текст, що міститься в цьому файлі. Крім цього, можуть ще бути присутнім посилання на файли з графічними зображеннями, які будуть відображатися разом з текстом при перегляді.

Для формування файлу HTML немає необхідності у використанні потужного текстового процесора, більш зручне використання, наприклад, редактора Notepad (Блокнот). Вміст HTML-сторінок представлено в гіпертекстовому вигляді, що передбачає наявність в документі, який відображається на екрані, виділених певним чином місць, клацнувши мишею по яким, можна переміститися в іншу частину цієї ж сторінки або на іншу сторінку.

MySQL

SQL (від Structured Query Language - структурована мова запитів) - створений для роботи з реляційними базами даних. Він дозволяє користувачам взаємодіяти з базами даних (переглядати, шукати, додавати, керувати даними). MySQL - розрахований на багато користувачів, багато-сервер бази даних SQL. Має хорошу швидкість і гнучкість, якщо використовувати його для зберігання зображень і файлів. MySQL відповідає специфікації ANSI 92 SQL.

Його переваги:

- підтримка декількох одночасних запитів (многопоточність);
- можливість запису фіксованою, а також змінної довжини;
- оптимізація зв'язків з приєднанням багатьох даних за один прохід;
- гнучка система паролів та доступів;
- ODBC драйвер в комплекті з ісходником;
- дані зберігаються в форматі ISO8859_1;
- інтерфейс з мовами C і perl;

- легко управляти таблицею (включаючи додавання і видалення ключів і полів);
- до 16 ключів в таблиці (в кожному ключі до 15 полів);
- підтримка ключових полів, а також спеціальних полів в операторі CREATE;
- підтримка чисел довжиною від 1 до 4 байт (ints, float, double, fixed), рядків змінної довжини і міток часу;
- утиліта перевірки і ремонту таблиці (isamchk);
- швидка система пам'яті, заснована на потоках;
- при роботі з рядками реєстр символів в оброблюваних рядках ролі не грає;
- псевдоніми застосовні не тільки до таблиць, але також до окремих колонок у таблиці;
- всі поля мають значення за замовчуванням. INSERT можна використовувати на будь-якому підмножині полів.

Основні переваги MySQL - швидкість, стійкість і легкість у використанні.

Висновки до розділу 2

У розділі 2 було проведено аналіз засобів розробки програмного забезпечення. Було розглянуто основні етапи розробки вебсайтів та його подальшого обслуговування.

Також були проаналізовані мови програмування за допомогою яких буде побудований вебсайт.

3 АРХІТЕКТУРА, МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Макет користувачького інтерфейсу

Будь-який проєкт завжди складається з двох рівнозначних частин, а саме:

- користувачький інтерфейс;
- програмно-апаратна частина.

Якщо хоча б один з цих пунктів спроектовано з помилками, або не виконує поставлені задачі, то шанс того, що проєкт буде комерційно невдалим досить великий. Що ж таке «інтерфейс» і з яких частин він складається?

Інтерфейс - спільний кордон між двома функціональними об'єктами, вимоги до якої визначаються стандартом; сукупність засобів, методів і правил взаємодії (управління, контролю і т. д.) між елементами системи.

Під сукупністю засобів і методів інтерфейсу користувача маються на увазі:

а) Засоби:

- виведення інформації з пристрою до користувача - весь доступний діапазон впливів на організм людини (зорових, слухових, тактильних, нюхових і т. д.) - екрани (дисплеї, проєктори) і лампочки, динаміки, зумери і сирени, вібромотор і т. д.;

- введення інформації / команд користувачем в пристрій – безліч всіляких пристроїв для контролю стану людини - кнопки, перемикачі, потенціометри, датчики положення і руху, сервоприводи, жести руками, навіть знімання мозкової активності користувача.

б) методи:

- набір правил, закладених розробником пристрою, згідно з якими сукупність дій користувача повинна привести до необхідної реакції пристрою і виконання необхідного завдання - так званий логічний інтерфейс.

Створення користувацького інтерфейсу складається з декількох етапів, основним з яких є створення мокапу інформаційної системи.

Етап створення мокапу дозволяє швидко зрозуміти бачення клієнта і зробити безліч змін до початку розробки інтерфейсу користувача. На даному кроці розташовуємо кнопки, форм і інші необхідні елементи, а вже потім підбираємо палітру кольорів, шрифти, зображення, перетворюючи все це в зручний і красивий макет. Тобто починаємо з варфрейма (план розташування елементів на сторінці), а закінчуємо створенням з цього плану красивої картинки.

Створюючи макет для вебзастосунку, ключове місце займає домашня сторінка інформаційної системи.



Рисунок 3.1 – мокап головної сторінки

Сторінка реєстрації – це сторінка, на яку потрапляє користувач, якщо він незареєстрований у системі (рис. 3.2). На ній буде відбуватися авторизація користувача. Дана форма матиме декілька обов’язкових полів для заповнення, а саме: повне ім’я, ім’я користувача, пароль, підтвердження паролю, електронна пошта та номер телефону. Останні два поля розраховані для контакту користувачів

та офісу підтримки, якщо у користувача виникнуть проблеми з програмним забезпеченням.

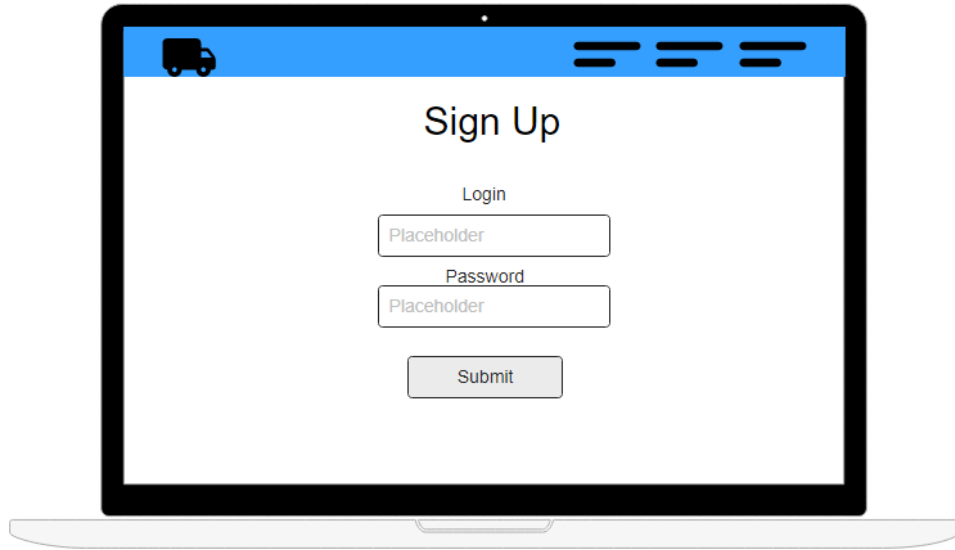


Рисунок 3.2 – мокап сторінки реєстрації

Вебзастосунок обов'язково повинен мати контактну форму для зворотнього зв'язку з власниками магазину для збільшення конверсії користувачів вебсайту:

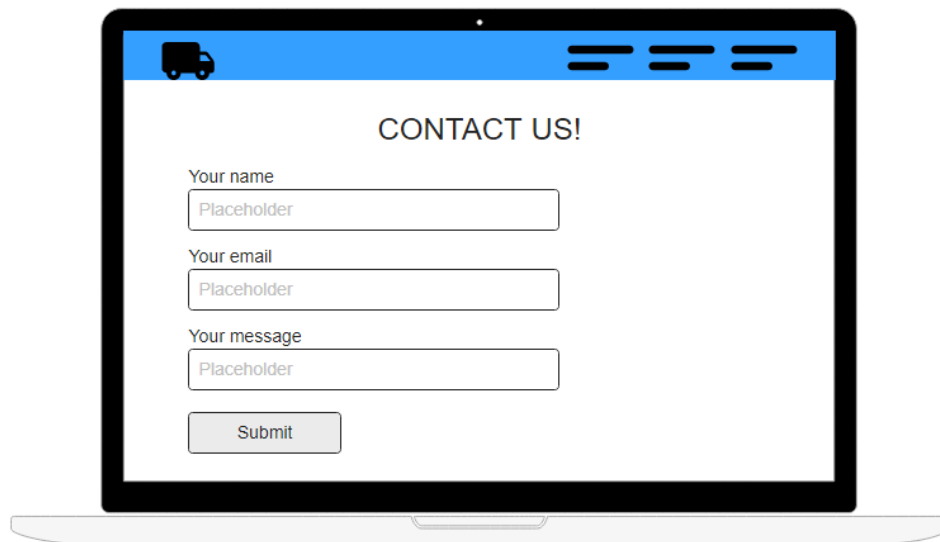


Рисунок 3.3- мокап форми зворотнього зв'язку

3.2 Інформаційна модель

Процеси управління, в яких на основі отриманої інформації необхідно приймати відповідні рішення, є найбільш очевидними для застосування методів моделювання. Найчастіше моделювання використовують для дослідження існуючої системи в разі, коли реальний експеримент проводити недоцільно через значні фінансові або трудових витрат, або при необхідності проведення аналізу проєктованої системи, яка ще фізично не існує в даній організації. Людина за допомогою інформаційної моделі отримує інформацію, на основі якої він може сформулювати образ реальної обстановки.

Процес побудови моделі є творчою процедурою, яка важко піддається формалізації. Модельні уявлення - це абстрактні образи елементів системи (об'єктів, технічних засобів, програмного забезпечення і т.д.). Їх сукупність дозволяє отримати досить повне уявлення про систему, яка створюється.

Однозначного поняття системи не існує. У загальному вигляді система представлена сукупністю взаємопов'язаних елементів, які утворюють певну цілісність, єдність. Від ступеня деталізації опису станів і умов функціонування об'єкта управління залежить кількість груп елементів інформаційної моделі. Найчастіше, елемент інформаційної моделі пов'язаний з яким-небудь параметром об'єкта управління. Режим перегляду даних і їх зв'язків називається моделлю даних. Розрізняють моделі ієрархічних, мережевих і реляційних даних, які, як правило, входять до складу систем управління базами даних (СКБД). В СУБД реалізуються моделі процесів накопичення і застосування інформації і знань. – Для формування моделі використовується:

- структурна схема об'єкта, який підлягає автоматизації;
- структурно-функціональна схема об'єкта, який підлягає автоматизації;
- алгоритм функціонування системи;
- розташування технічних засобів на об'єкті у вигляді схеми;

3.3 Сценарії використання

Сценарії використання- це письмовий опис того, як користувачі виконуватимуть завдання на вебсайті. Він описує, з точки зору користувача, поведінку системи, коли вона відповідає на запит. Кожен варіант використання представлений у вигляді послідовності простих кроків, починаючи з мети користувача і закінчуючи досягненням цієї мети.

Сценарії використання додають цінності, оскільки вони допомагають пояснити, як повинна вести себе система, і в процесі вони також допомагають зрозуміти, що може піти не так. Вони містять перелік цілей, і цей список можна використовувати для встановлення вартості та складності системи. Потім команди проекту можуть домовитися про те, які функції стають вимогами та будуються.

Таблиця 2.1 – Характеристика сценаріїв використання

Сценарії використання показують:	Сценарії використання не показують:
Хто користується веб-сайтом	Мову реалізації
Мету користувача	Інформація про інтерфейси
Кроки, які виконує користувач для виконання певного завдання	
Відповідь вебсайта на дію користувача	

Таблиця 2.2 – Опис прецеденту «Реєстрація у системі»

Use case section	Comment
Usecase Name	Реєстрація у системі
Scope	Вебзастосунок магазину автотоварів
Level	Успішно пройти реєстрацію і зареєструватися в системі
Primary Actor	Користувач

Кінець таблиці 2.2

Use case section	Comment
Stakeholders and interests	Користувач- пошук необхідної інформації
Preconditions	Користувач перейшов на сторінку реєстрації
Success guarantee	1. Користувач використовує браузер з підтримкою HTML 5, CSS 3; 2. Користувач має використовувати стабільне підключення
Main Success Scenario	Користувач реєструється у системі
Extensions	1. Користувач залишає порожніми основні поля: – користувач переходить на сторінку реєстрації; – користувач залишає порожнім обов'язкове поле; – користувач натискає кнопку реєстрації; – система видає повідомлення про порожнє поле. 2. Користувач заповнює поля без дотримання вимогам: – користувач переходить на сторінку реєстрації; – користувач заповнює поле без дотримання вимогам; – користувач натискає кнопку реєстрації; – система видає повідомлення про некоректність форми заповнення полей.
Special Requirements	Швидкість доступу до мережі Інтернет має бути більшою ніж 60 кб/сек.
Frequency of Occurrence	< 10%
Miscellaneous	Чи необхідно підтверджувати поштову скриньку

Таблиця 2.3 – Опис прецеденту «Зворотній зв'язок»

Use case section	Comment
Usecase Name	Зворотній зв'язок
Scope	Вебзастосунок магазину автотоварів
Level	Успішно відправити лист за допомогою форми
Primary Actor	Користувач

Кінець таблиці 2.3

Use case section	Comment
Stakeholders and interests	Користувач- відправка листа
Preconditions	Користувач перейшов на головну сторінку і зареєструвався в системі
Success guarantee	1. Користувач використовує браузер з підтримкою HTML 5, CSS 3; 2. Користувач має використовувати стабільне підключення
Main Success Scenario	Користувач відправляє лист
Extensions	Користувач залишає порожніми основні поля: <ul style="list-style-type: none"> – користувач переходить на головну сторінку; – користувач залишає порожнім обов'язкове поле; – користувач натискає кнопку «виправити»; – система видає повідомлення про порожнє поле.
Special Requirements	Швидкість доступу до мережі Інтернет має бути більшою ніж 60 кб/сек.
Frequency of Occurrence	< 10%

Таблиця 2.4 – Опис прецеденту «Запит до бота»

Use case section	Comment
Usecase Name	Запит до бота
Scope	Вебзастосунок магазину автотоварів
Level	Успішно зробити запит до бота і отримати відповідь
Primary Actor	Користувач
Stakeholders and interests	Користувач- здобуття інформації

Кінець таблиці 2.4

Use case section	Comment
Preconditions	Користувач перейшов на головну сторінку і зареєструвався в системі
Success guarantee	1. Користувач використовує браузер з підтримкою HTML 5, CSS 3; 2. Користувач має використовувати стабільне підключення
Main Success Scenario	Користувач здобуває необхідну інформацію
Extensions	1. Користувач відправляє некоректний запит <ul style="list-style-type: none"> – користувач переходить на головну сторінку; – користувач відправляє некоректний запит; – користувач натискає кнопку «виправити»; – система видає повідомлення про помилку. 2. Система не знає відповіді на запитання <ul style="list-style-type: none"> – користувач переходить на головну сторінку; – користувач відправляє запит; – користувач натискає кнопку «виправити»; – система видає повідомлення про помилку.
Special Requirements	Швидкість доступу до мережі Інтернет має бути більшою ніж 60 кб/сек.
Frequency of Occurrence	< 10%

Сценарії використання- це письмовий опис того, як користувачі виконуватимуть завдання на вебсайті. Він описує, з точки зору користувача, поведінку системи, коли вона відповідає на запит. Кожен варіант використання представлений у вигляді послідовності простих кроків, починаючи з мети користувача і закінчуючи досягненням цієї мети.

Сценарії використання додають цінності, оскільки вони допомагають пояснити, як повинна вести себе система, і в процесі вони також допомагають зрозуміти, що може піти не так. Вони містять перелік цілей, і цей список можна

використовувати для встановлення вартості та складності системи. Потім команди проекту можуть домовитися про те, які функції стають вимогами та будуються.

Проектована система представляється у виді множини сутностей чи акторів, взаємодіючих з системою за допомогою прецедентів. При цьому актором чи діючою особою є будь-яка сутність, взаємодіюча з системою зовні. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система.

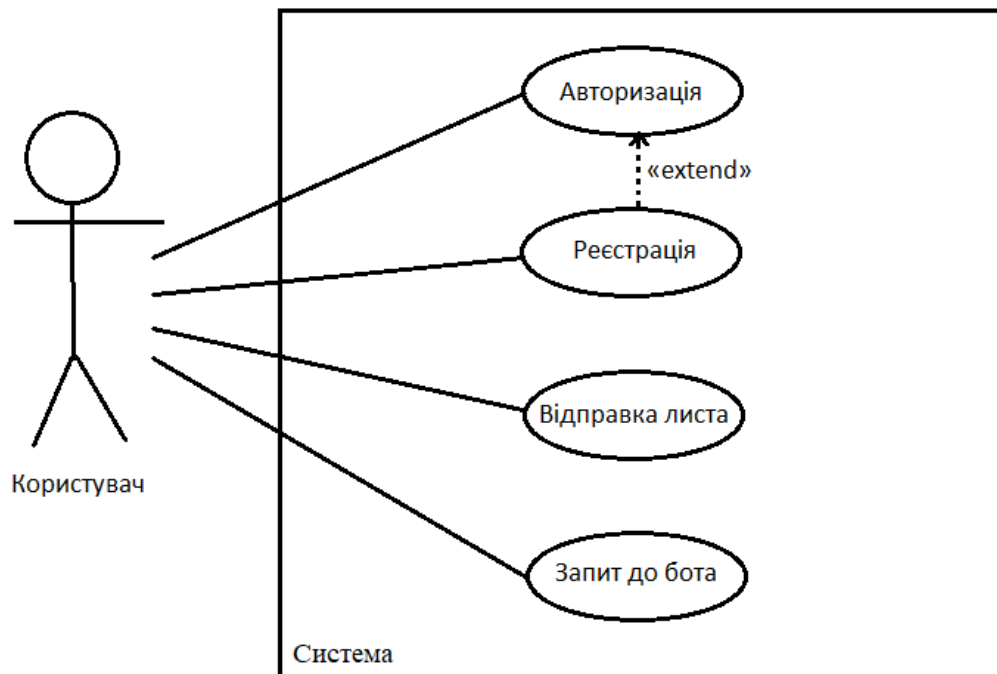


Рисунок 3.4 – Діаграма використання

На рисунку 3.4 представлена діаграма використання розроблюваного програмного забезпечення.

Висновки до розділу 3

Важливим етапом при розробці користувацького програмного забезпечення є створення макетів його зовнішнього виду та інтерфейсу користувача. Протягом розділу створено мокапи основних сторінок вебзастосунку.

На основі проведеного аналізу було виявлено основні функції вебзастосунку і сценарії їх використання.

4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Підготовка і створення HTML-сторінки.

Для початку роботи над інтернет-сторінкою треба встановити та налаштувати локальний сервер OpenServer.

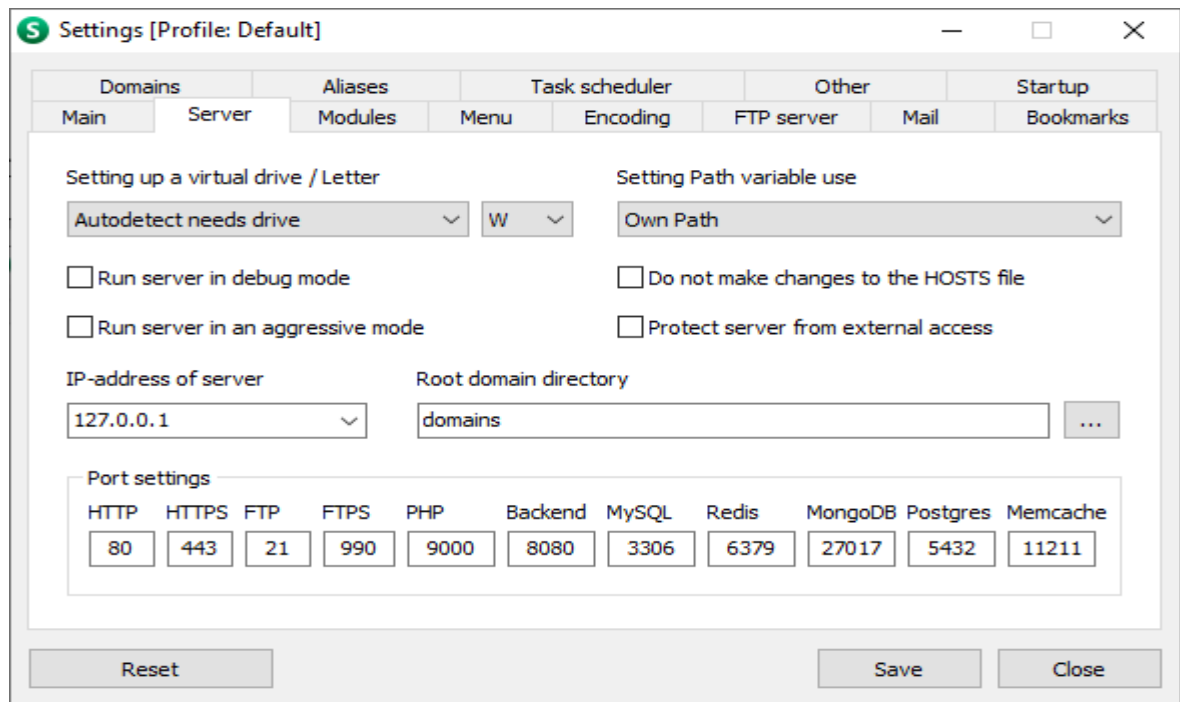


Рисунок 4.1 - налаштування OpenServer

Першим кроком у розробці вебсайту є створення його основи, яка полягає в реалізації простого HTML-коду. Цей крок є дуже важливим так як від нього залежить конструкція кінцевого варіанту вебзастосунку. Об'єм текстового контенту в кожному блоці слід вирішити заздалегідь для загального розуміння відношення розмірів блоків один до одного.

Перед початком розміщення контенту на вебсайт слід звернути увагу на його оформлення. За допомогою таблиці стилів CSS треба описати зовнішній вигляд для всієї сторінки.



Рисунок 4.2 - зображення головної сторінки

Після побудови конструкції вебсайту і задання стилів для нього можна починати наповнювати сторінку необхідними зображеннями та текстом.

4.2 Наповнення сторінки та адаптивність

Для того щоб сторінка не здавалася пустою, окрім простого тексту та зображень треба додати header, на якому розміщується основна інформація (графік роботи, адреса, тощо) і навігація по сторінці, footer, який показує де сторінка завершується, та різні інтерактивні іконки, які можуть виділятися на фоні решти сторінки.

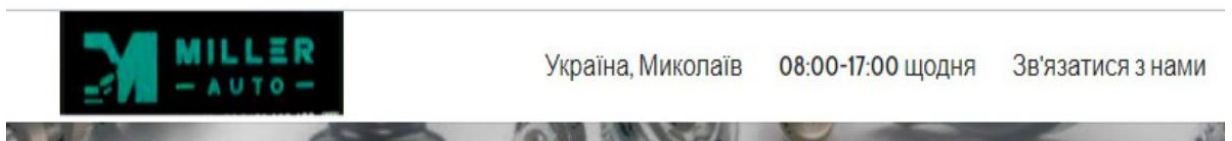


Рисунок 4.3 – header сторінки



Рисунок 4.4 – інтерактивні іконки

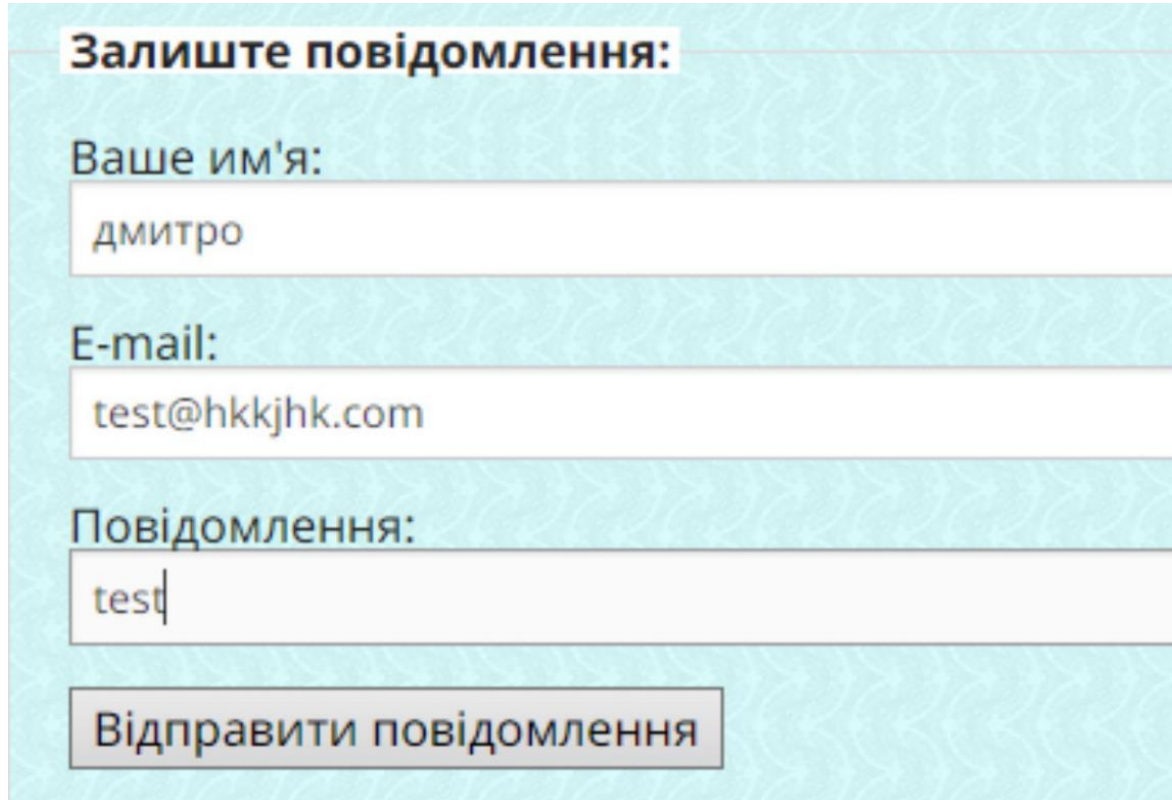


Рисунок 4.5 – випадаюче меню

Для того, щоб header сайту, і вся сторінка взагалом виглядала доцільно на менших екранах, у ній було реалізовано випадаюче меню.

4.3 Зворотній зв'язок

На випадок, коли користувачу треба буде дізнатися інформацію, якої немає на сайті, або користувач бажає замовити послуги, які надає власник сторінки, на ній було реалізовано форму зворотного зв'язку, яка передає усі повідомлення в базу даних.



Залиште повідомлення:

Ваше ім'я:
дмитро

E-mail:
test@hkkjkh.com

Повідомлення:
test

Відправити повідомлення

Рисунок 4.6 – форма зворотного зв'язку

Реалізація форми:

```
<form action="mail.php" method="post" >
  <fieldset>
    <legend>Залиште повідомлення:</legend>
    Ваше ім'я:
    <input type="text" name="namee" size=20>
    E-mail:
    <input type="text" name="email1">
    Повідомлення:
    <input type="text" name="message">
    <input type="submit" value="Відправити повідомлення">
  </fieldset>
</form>
```

Рисунок 4.7 – реалізація контактної форми

```
<?php
$namee = $_POST["namee"];
$emaill = $_POST["emaill"];
$messagee = $_POST["messagee"];
$namee = htmlspecialchars($namee);
$emaill = htmlspecialchars($emaill);
$messagee = htmlspecialchars($messagee);
$mysqli = new mysqli('localhost', 'root',
'', 'mysql');
$mysqli->query("INSERT INTO `messages` (`id`
`, `user`, `email`, `user_message`)
VALUES (NULL, '$namee', '$emaill', '$
messagee')");
?>
```

Рисунок 4.8 – підключення бази даних

	id	user	email	user_message
<input type="checkbox"/>	7	asd	asdasd	
<input type="checkbox"/>	11	sdfasdfas	asdfasdfas	sadfasdf
<input type="checkbox"/>	12	Дмитрий	dima@gmail.com	кваліфікаційна робота
<input type="checkbox"/>	13	дмитро	test@hkkjhk.com	test

Рисунок 4.9 – база даних форми

Повідомлення відправлене з контактної форми буде зберігатись у базі даних, що можна побачити на рисунку 4.9.

4.3 Реєстрація та авторизація

З метою розширення клієнтської бази до сторінки було добавлено можливість реєструватися та авторизовуватись.

Реєстрація

Ваш логін:

Ваш пароль:

Рисунок 4.10 – форма для реєстрації

Адреса `save_user.php` - це адреса обробника. Тобто, після натискання на кнопку "Зареєструватися", дані з полів відправляються на сторінку `save_user.php` методом "post". У текстове поле (`name = "login" type = "text"`) користувач вводить свій логін. В поле для паролів (`name = "password" type = "password"`) користувач вводить свій пароль. Кнопка (`type = "submit"`) відправляє дані на сторінку `save_user.php`.

```
<form action="save_user.php" method="post">
<p>
  <label>Ваш логін:<br></label>
  <input name="login" type="text" size="15" maxlength="15">
</p>
<p>
  <label>Ваш пароль:<br></label>
  <input name="password" type="password" size="15" maxlength="15">
</p>
<p>
  <input type="submit" name="submit" value="Зареєструватися">
</p></form>
<?php
$db = mysqli_connect("127.0.0.1", "root", "");
mysqli_select_db ($db,"mysql");
?>
```

Рисунок 4.11 – реалізація реєстрації

← T →				id	login	password
<input type="checkbox"/>				1	sdfsaf	asda
<input type="checkbox"/>				2	asdfas	sadfdasf
<input type="checkbox"/>				3	dima	1234
<input type="checkbox"/>				4	565674	567457
<input type="checkbox"/>				5	asdasd	asdsadas
<input type="checkbox"/>				6	1111	1111

Рисунок 4.12 – база даних зареєстрованих користувачів

Адреса `testreg.php` - це адреса обробника. Тобто, після натискання на кнопку "Увійти", дані з полів відправляються на сторінку `testreg.php`, в яка перевіряє співпадіння логіну та паролю з базою даних, методом "post". У текстове поле (`name = "login" type = "text"`) користувач вводить свій логін. В поле для паролів (`name = "password" type = "password"`) користувач вводить свій пароль. Кнопка (`type = "submit"`) відправляє дані на сторінку `testreg.php`. Залишається перевірити на порожність змінні логіна і ід користувача. Якщо порожні, то посилання не виводиться. Якщо не порожні, то навпаки.

Вхід у систему

Ваш логін:

Ваш пароль:

[Зареєструватися](#)

Рисунок 4.13 – форма для входу на сайт


```
</br></br></br>
<h2>Вход в систему</h2>
<p>
<label>Ваш логин:<br></label>
<input name="login" type="text" size="15" maxlength="15">
</p>
<p>
<label>Ваш пароль:<br></label>
<input name="password" type="password" size="15" maxlength="15">
</p>
<p>
<input type="submit" name="submit" value="Войти">
<br>
<a href="reg.php">Зарегистрироваться</a>
</p></form>
<br>
<?php
if (empty($_SESSION['login']) or empty($_SESSION['id']))
{
echo "Вы вошли на сайт, как гость<br><a href="#">Эта ссылка доступна только зарегистрированным пользователям</a>";
}
else
{
echo "Вы вошли на сайт, как ".$_SESSION['login']. "<br><a href='glav.html'>Нажмите чтобы пререйти на сайт</a>";
}
?>
</div>
</body>
</html>
```

Рисунок 4.14 – реалізація форми входу на сайт

Уся процедура авторизації працює на сесіях. Саме в ній зберігаються дані користувача, поки він знаходиться на сайті. Дуже важливо запустити їх на самому початку сторінки. Якщо змінна \$login порожня, то вона знищується. Якщо змінна \$password порожня, то вона знищується. У випадку, коли користувач не ввів логін або пароль, то помилка видаляється, а скрипт зупиняється.

```
1 <?php
2 session_start();
3 if (isset($_POST['login'])) { $login = $_POST['login']; if ($login == '') { unset($login); } }
4 if (isset($_POST['password'])) { $password = $_POST['password']; if ($password == '') { unset($password); } }
5
6 if (empty($login) or empty($password))
7     exit ("Вы ввели не всю информацию, вернитесь назад и заполните все поля!");
8 }
9 $login = stripslashes($login);
10 $login = htmlspecialchars($login);
11 $password = stripslashes($password);
12 $password = htmlspecialchars($password);
13 $login = trim($login);
14 $password = trim($password);
15 include ("bd.php");
16 $result = mysqli_query($db,"SELECT * FROM users WHERE login='$login'"); /
17 $myrow = mysqli_fetch_array($result);
18 if (empty($myrow['password']))
19 {
20
21     exit ("Извините, введенный вами login или пароль неверный.");
22 }
23 else {
24     if ($myrow['password']==$password) {
25         $_SESSION['login']=$myrow['login'];
26         $_SESSION['id']=$myrow['id'];
27         echo "Вы успешно вошли на сайт! <a href='index.php'>Главная страница</a>";
28     }
29     else {
30         exit ("Извините, введенный вами login или пароль неверный.");
31     }
32 }
```

Рисунок 4.15 – реалізація обробника даних

Якщо логін і пароль введені, то вони обробляються, щоб теги і скрипти не працювали. З бази витягуються всі дані про користувача з введеним логіном, і якщо паролі збігаються, то сесія запускається у користувача.

4.4 Створення чат-бота

Чат-бот — це програмна програма, яка може імітувати людські онлайн-розмови з користувачем.

Спочатку було створено каркас бота, який складається з трьох секцій: заголовка, основної частини та поля введення. Усі вони містяться в контейнері, який сам знаходиться всередині головної сторінки.

Область заголовка містить текст заголовка програми чату. Секція основної частини містить усі повідомлення як від користувача, так і від бота. Ці повідомлення будуть динамічно створюватися з бази даних і вставлятися на сторінку за допомогою JavaScript. Поле введення містить форму, яка визначає спосіб збору повідомлень користувача з інтерфейсу.

Після цього треба придати належний зовнішній вигляд для всієї секції бота.

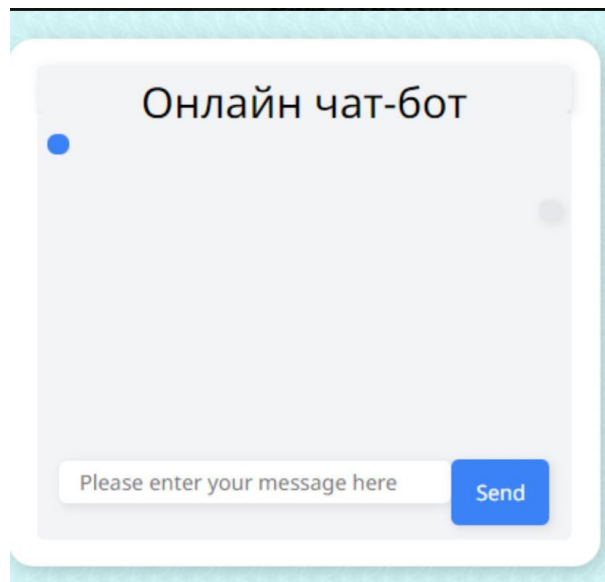


Рисунок 4.16 – зовнішній вигляд чат-боту

Для обробки запитів AJAX, підключення до бази даних і отримання належної репліки відповідає PHP-скрипт. Саме до цього скрипту треба робити запити POST AJAX з інтерфейсу.

```
*/
$conn = mysqli_connect("127.0.0.1","root","","bot");

if($conn)
{
    $user_messages = mysqli_real_escape_string($conn, $_POST['messageValue']);
    $query = "SELECT * FROM bot WHERE messages LIKE '%$user_messages%'";
    $makeQuery = mysqli_query($conn, $query);
    if(mysqli_num_rows($makeQuery) > 0)
    {
        $result = mysqli_fetch_assoc($makeQuery);
        echo $result['response'];
    }else{
        echo "Sorry, I can't understand you.";
    }
}else {
    echo "Connection failed" . mysqli_connect_errno();
}
```

Рисунок 4.17 – скрипт для обробки запитів AJAX

Наступним кроком є додавання повідомлень до бази даних за допомогою інструменту phpMyAdmin. У створеній базі даних знаходиться таблиця «bot», яка містить 3 поля:

- id (автоматичний інкремент): містить унікальний ідентифікатор для кожного запису;
- messages: це поле для можливих запитів користувачів;
- response: поле містить відповіді бота на відповідні запити користувача.

		id	messages	response
<input type="checkbox"/>		3	c	d
<input type="checkbox"/>		5	d	h
<input type="checkbox"/>		7	Привіт	Доброго дня!
<input type="checkbox"/>		8	Який у вас графік роботи?	Ми працюємо щодня з 8:00 до 17:00
<input type="checkbox"/>		9	Скільки коштує доставка?	Вартість залежить від вашого місця проживання. Під...

Рисунок 4.18 – база даних чат-боту

З усіма даними, наявними в базі даних, і написаним PHP-скриптом, останнім кроком буде запит AJAX з інтерфейсу та вставка нових повідомлень як від користувача, так і від бота, за допомогою JavaScript.

```
document.querySelector("#send").addEventListener("click", async () => {
  let xhr = new XMLHttpRequest();
  var userMessage = document.querySelector("#userInput").value

  let userHtml = '<div class="userSection">' + '<div class="messages user-message">' + userM
  '<div class="seperator"></div>' + '</div>'

  document.querySelector("#body").innerHTML += userHtml;
  xhr.open("POST", "query.php");
  xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
  xhr.send(`messageValue=${userMessage}`);

  xhr.onload = function () {
    let botHtml = '<div class="botSection">' + '<div class="messages bot-reply">' + this.r
    '<div class="seperator"></div>' + '</div>'
  }
}
```

Рисунок 4.19 – запит AJAX з інтерфейсу

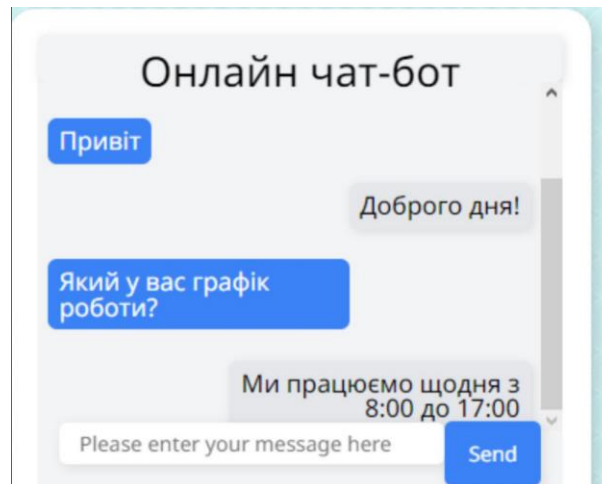


Рисунок 4.20 – працюючий бот

Тепер, при натисканні кнопки відправити, повідомлення користувача спочатку витягується та вставляється на сторінку. Потім до сценарію сервера надсилається запит AJAX, щоб отримати відповідь.

Висновки до розділу 4

За допомогою попередньо освітлених засобів розробки у процесі виконання практичної частини було створено вебзастосунок. Було додано можливість реєстрації та авторизації для користувачів. Для полегшення знаходження необхідної інформації, на вебсайті інтегровано чатбота-помічника та форму зворотнього зв'язку.

ВИСНОВКИ

У сучасному світі автозапчастини є такими ж безальтернативними товарами, як і продукти харчування чи ліки. Попит всередині цієї сфери не зникне ніколи. Вебсайт для існуючого магазину з продажу автозапчастин – один із найперспективніших напрямків для поширення бізнесу. Адже за допомогою Інтернету можна з мінімальними витратами часу відшукати оригінальні запчастини як для сучасних, так і знятих з виробництва моделей автомобілів.

У ході проведення роботи було вирішено поставлені завдання.

Результатом кваліфікаційної роботи бакалавра став готовий до роботи вебзастосунок, в якому розміщена інформація про офлайн-магазин автозапчастин. Під час виконання було досягнуто поставленої мети, а саме покращення зворотного зв'язку між користувачем та власником магазину. Мету було здійснено шляхом додавання до вебсайту контактної форми для залишення повідомлень, та інтеграції чат-боту, який має змогу допомогти користувачу у вирішенні незначних питань без втручання адміністрації магазину.

Надалі розроблений вебсайт можна буде використовувати у реальному житті. Також можливе розширення списку послуг, акцій, новин. Даний ресурс стабільний у роботі та не вимагає високих знань для роботи з ним.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Culwell L. Improve Your Website, Improve Your Business. Bonus Books, 2007. 175 p.
2. Lawson B. Introducing HTML 5. New Riders Press, 2010.
3. Чи потрібен сайт для бізнесу? Корпоративний сайт компанії - Блог. *Блог*. URL: <https://webtime.com.ua/sayt-dlia-biznesu/> (дата звернення: 15.03.2022).
4. Веб-сайт: визначення й застосування. Статті про розробку та просування веб-сайтів. *Разработка веб-сайтов Харьков, раскрутка веб-сайтов, веб-дизайн, Украина, Харьков - студия WebTec*. URL: <http://www.webtec.com.ua/uk/articles/index/view/2011-05-05/web-site> (дата звернення: 16.04.2022).
5. Brown T. B. CSS Master. SitePoint, 2018. 354 p.
6. PHP Tutorial. *W3Schools Online Web Tutorials*. URL: <https://www.w3schools.com/php/> (date of access: 04.05.2022).
7. Специфікація вимог до програмного забезпечення. *www.wiki.uk-ua.nina.az*. URL: https://www.wiki.uk-ua.nina.az/Специфікація_вимог_до_програмного_забезпечення.html (дата звернення: 17.05.2022).
8. Mayer G. E. T., Awesomeness J. JavaScript: JavaScript Awesomeness Book. CreateSpace Independent Publishing Platform, 2017. 68 p.
9. Darlington K. Effective Website Development: Tools and Techniques. Addison Wesley Publishing Company, 2005. 476 p.
10. Bittner K. Use case modeling. Boston, MA : Addison-Wesley, 2003. 347 p.
11. Різниця між SQL vs MySQL - Luxnet - Luxnet.io. *Ukrainian Software Development Outsourcing Provider - Luxnet.io*. URL: <https://luxnet.io/uk/blog/sql-vs-mysql-how-to-differentiate> (дата звернення: 14.06.2022).

12. CSS: Cascading Style Sheets | MDN. *MDN Web Docs*.
URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (date of access: 24.06.2022).
13. Етапи створення веб сайтів: які є основні кроки розробки -
WebTune. *Webtune*. URL: <https://webtune.com.ua/statti/web-rozrobka/etapy-stvorennya-veb-sajtiv/> (дата звернення: 16.06.2022).
14. HTML basics - Learn web development | MDN. *MDN Web Docs*.
URL: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics (date of access: 15.06.2022).
15. Переваги MySQL | Портал знань, портал знаний, дистанційне навчання. *Портал знань – Знання повинні бути доступними | Портал знань, портал знаний, дистанційне навчання*.
URL: <http://www.znannya.org/?view=concept:304> (дата звернення: 03.06.2022).
16. Вебсайти, створення, оновлення, оптимізація - Zona Digital - Digital Agency - Web Development. *Home - Zona Digital - Digital Agency - Web Development*.
URL: <https://zd.com.ua/websites> (дата звернення: 19.06.2022).
17. Переваги та недоліки JavaScript. *hackit-ukraine*. URL: <https://hackit-ukraine.com/627-the-advantages-and-disadvantages-of-javascript> (дата звернення: 16.06.2022).
18. *mynizhyn.com*. Переваги, використання та вивчення JavaScript. *Головний сайт Ніжина, де ви знайдете всю інформацію про новини, події та історію міста*.
URL: <https://mynizhyn.com/news/ukraina-i-svit/17122-perevagi-vikoristannja-ta-vivchennja-javascript.html> (дата звернення: 22.06.2022).
19. ІНФОРМАЦІЙНА МОДЕЛЬ ОБ'ЄКТА. *Stud*.
URL: https://stud.com.ua/97318/informatika/informatsiyna_model_obyekta (дата звернення: 24.06.2022).