

Сумський державний педагогічний університет імені А. С. Макаренка

Фізико-математичний факультет

Кафедра інформатики

Цюх Антон Олегович

**МЕТОДИ МОНІТОРИНГУ ВЕБ-РЕСУРСІВ
(НА ПРИКЛАДІ РОЗРОБКИ ІНТЕРНЕТ-МАГАЗИНУ)**

Спеціальність 122 Комп'ютерні науки

Галузь знань: 12 Інформаційні технології

Кваліфікаційна робота

на здобуття освітнього ступеня бакалавр

Науковий керівник:

_____ О.Г.Медведовська,
кандидат фізико-математичних наук,
доцент

_____ 2021 року

Виконавець:

_____ А.О.Цюх

_____ 2021 року

Суми-2021

Зміст

ВСТУП	3
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1 Методи моніторингу веб-ресурсів	6
1.2 Інтернет-магазини як поширені веб-ресурси	8
1.3 Аналіз аналогів інтернет-магазинів.....	10
РОЗДІЛ 2 РОЗРОБЛЕННЯ ВЕБ-РЕСУРСУ ТА ВИКОРИСТАННЯ МЕТОДІВ МОНІТОРИНГУ В НЬОМУ	15
2.1 Визначення структури додатку	15
2.2 Розробка інтерфейсу користувача.....	18
2.3 Розробка функціоналу сайту.....	26
2.4 Тестування інтернет-магазину.....	29
ВИСНОВКИ.....	33
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	34
Додаток А.....	36

ВСТУП

Бізнес в інтернеті зараз сильно зав'язаний не тільки на якість самого сайту, але і на його доступність і працездатність в режимі 24/7. І хоча хостинг-провайдери заявляють на досить великому аптаймі (час безперервної роботи) частини майданчиків і серверів, не завжди цього буває достатньо, щоб бути впевненим у надійності сайту. Одним із помічників в цьому є моніторинг як доступності, так і працездатності основних підсистем сайту.

Веб-аналітика - система вимірювання, збору, аналізу, представлення та інтерпретації інформації про відвідувачів веб-сайтів з метою їх покращення та оптимізації. Головним завданням веб-аналітики є моніторинг відвідуваності веб-сайтів, на підставі даних якого визначається кількість відвідувачів та їх поведінка для прийняття рішень з розвитку і розширенню функціональних можливостей веб-ресурсу. Веб-аналітика дозволяє не тільки працювати над поліпшенням сайтів, але і проводити роботи по оптимізації бюджету на онлайн-просування [8].

У 1990 році разом з народженням HTTP-протоколу почалася ера веб-аналітики. Став можливим запис взаємодії користувачів і сервера в лог-файли. Кожен раз, коли користувач Інтернету викликає HTML-елемент, в лог-файл записується рядок - хіт. З ростом відвідуваності власники сайтів стали отримувати дуже багато хітів, лог-файли збільшувалися в розмірах. Це мав бути спосіб аналізу таких обсягів даних. Початок комерційної веб-аналітики можна вважати створенням компанії WebTrends в 1993 році. У 1995 році була створена система Analog - перша безкоштовна система аналізу лог-файлів. Analog дозволяв генерувати звіти з лог-файлів, має зрозумілою документацію і можливість графічної інтерпретації даних. Нарешті веб-аналітикою змогли займатися не тільки програмісти, але і професійні маркетологи. Згодом сторінки стали містити більшу кількість елементів - завантаження якогось елементу перестала означати завантаження сторінки. Виник новий метод збору інформації про відвідування сторінок - javascript-теги, що став найбільш

поширеним з розвитком мереж. Javascript-теги впроваджуються власником на всі сторінки сайту. При завантаженні сторінки виконується тег - збирає інформацію про візит користувача і зберігає в базу даних. У 2005 році Google створила свою систему веб-аналітики Google Analytics, купивши компанію Urchin Software Corporation і її проект Urchin. У 2006 році стартував аналіз поведінки відвідувачів на сторінці. Стали вестися записи поведінки відвідувача на сторінці. Формуватися карти кліків і скролінгу. Веб-аналіз перестав бути тільки кількісним, але і став якісним.

Об'єкт: веб-ресурси та їх аналіз

Предмет: методи моніторингу веб-ресурсів (на прикладі розробки інтерфейсу інтернет-магазину).

Мета: описати методи моніторингу веб-ресурсів і на прикладі розробки інтерфейсу інтернет-магазину покати їх реалізацію.

Досягнення мети потребувало виконання низки **завдань**:

- 1) уточнити основні поняття дослідження;
- 2) охарактеризувати методи моніторингу веб-ресурсів;
- 3) розробити веб-сторінку інтернет-магазину;
- 4) продемонструвати реалізацію методів моніторингу на розробленому веб-ресурсі.

У процесі виконання роботи було використано низку **методів**:

теоретичні – термінологічний аналіз для уточнення понять дослідження; контент-аналіз для сторінок інтернет-магазинів для розроблення власного веб-ресурсу подібних;

емпіричні – опитування та бесіди з фахівцями щодо підбору кольорових рішень, наповнюваності, проведення аналізу веб-ресурсу.

Практичне значення роботи полягає у розробленості веб-сторінки Сумського м'ясокомбінату з урахуванням методів моніторингу веб-ресурсу [4].

Робота складається зі вступу, двох розділів, висновків, списку використаних джерел і додатку.

У першому розділі описано методи моніторингу веб-ресурсів.

Другий розділ присвячено розробленню авторського веб-ресурсу з використанням методів моніторингу.

РОЗДІЛ 1

ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Методи моніторингу веб-ресурсів

Моніторинг Інтернету - це комплекс операцій з пошуку, аналізу, класифікації, кластеризації інформації з Інтернету за тематикою, що цікавить користувача [1]. Таким чином поняття «моніторинг Інтернету» відноситься до моніторингу інтернет-ресурсів, на відміну від поняття «моніторинг мережі», яке відноситься до задач управління мережею. Моніторинг Інтернету дозволяє відстежувати і спостерігати за розвитком тієї або іншої події або теми. З деяких пір моніторинг інтернет-ресурсів став однією з ключових складових діяльності маркетингових служб, інформаційних агентств і служб новин [8].

Існують такі основні методи та інструменти:

- обробка вибраних сайтів у ручному режимі
- пошукові системи і каталоги
- новинна підписка (зокрема RSS)
- спеціалізовані сервіси і програмні продукти моніторингу Інтернету [7].

Регулярна перевірка. Один з найбільш поширених методів - регулярна перевірка працездатності сайту. Можна робити самому - відкрити з ранку сайт улюбленої компанії або інтернет-магазин (маючи його в стартовій сторінці браузера), радіти, що працює. І закривати. Не дуже зручно і зовсім не ефективно. Проблеми з доступністю зазвичай починаються непередбачувано. Виявляти їх краще відразу ж, як тільки вони виникли. Також часто допомагають самі користувачі - починають дзвонити і писати, що сайт не працює (якщо телефон і пошта вказані десь за межами сайту). Самий же частий сигнал неробочого сайту - припиняють приходити клієнти.

Моніторинг доступності. Більш освічені й просунуті власники сайту або керівники, за нього відповідають, знають про необхідність постійного моніторингу доступності сайту - відстеження того, чи працює він чи ні. Найбільш часта помилка тут - неправильний вибір частоти моніторингу.

Перевірка раз на добу працездатності сайту, звичайно, корисна, але втрата денної виручки - це занадто велика ціна помилки. Найбільш оптимальним для простого моніторингу є десяти хвилинний інтервал: більшість користувачів спробують повернутися на сайт протягом 1-2 годин, а за цей час можна як виявити проблеми, так і ефективно їх усунути без особливої шкоди для бізнесу компанії. З іншого боку, більш часті перевірки не дають гарантії того, що проблема буде виправлена оперативніше, ніж протягом години. Адже основний час йде не стільки на знаходження проблеми, скільки на з'ясування її причин та їх виправлення.

Моніторинг проблем. Коли рівень доступності сайту вже стає критичним для бізнесу - особливо наявність «плаваючих» проблем, пов'язаних зі збігом низки обставин - то звичайного моніторингу стає мало. Тут необхідний вже відстежувати кілька параметрів сайту з частотою не рідше ніж раз на хвилину і з кількох географічних точок (щоб максимально покрити хвилинний інтервал перевірок і встановити можливі проблеми, пов'язані з географією користувачів).

Серед можливих критеріїв перевірки можна виділити:

Проблеми з DNS-сервером (коли в певні інтервали часу адреса сайту не може бути визначений, хоча сам сайт фізично доступний).

Проблеми з великим часом відповіді (при оновленні кешу, наприклад, або при виконанні «важких» завдань на стороні сервера).

Проблеми з плановим виконанням завдань (в результаті яких сайт буде не доступний тільки в певні моменти часу).

Проблеми з великим часом очікування статичних файлів (наприклад, через мережевий інфраструктури або проблем з фізичним носієм).

Проблеми з підключенням до бази даних. і так далі. У цьому випадку необхідний щохвилинний моніторинг сайту з декількох точок (або за допомогою декількох незалежних сервісів або точок перевірки).

Моніторинг може бути недовготривалим (до виявлення і виправлення проблем) або періодичним (з метою профілактики проблем). Також можлива

регулярна самоперевірка сайту на можливі проблеми доступності, але зовнішня перевірка завжди краще внутрішньої за великим рядом причин. Додатково зовнішні сервіси вже зараз надають детальну інформацію про проблеми, аж до логів помилок на стороні клієнта (при відповідній настройці - і логів помилок з боку сервера). Цей метод особливо гарний, коли потрібно відловити якусь «плаваючу» помилку. При включенні детальних логів виникає помилки навіть при відсутності деталей помилки на стороні сервера можна її ефективно відстежити і усунути. Кілька точок перевірки дозволяють домогтися частоти перевірки аж до разу на 10 секунд - а це більш ніж достатньо, щоб виявити все, що необхідно.

Моніторинг працездатності

Ця тема особливо актуальна для інтернет-магазинів, для яких можливість замовлення або відправки запиту є ключовою. Також сюди може ставитися будь-який складний функціонал, яким можемо бути порушено змінами на сайті (наприклад, кабінет інтернет-банку). В цьому випадку необхідно налаштовувати ланцюжка перевірок або ставити складні умови для проведення перевірок. Без допомоги кваліфікованого фахівця тут навряд чи обійтися, але зараз є продукти і сервіси, які дозволяють автоматизувати весь або майже весь процес такого функціонального моніторингу сайту.

Наскільки важливий моніторинг для сайту оцінити дуже просто: достатньо поррахувати денний дохід, створюваний сайтом, і співвіднести його з вартістю цього моніторингу. Зрозуміти, скільки коштує кожна година простою сайту в робочий час, і прийняти рішення про ефективність моніторингу та його деталізації (буде це проста перевірка доступності, моніторинг працездатності або комплексний моніторинг всіх можливих проблем).

1.2 Інтернет-магазини як поширені веб-ресурси

Інтернет-магазин — це програмний комплекс, який дозволяє вести комерційну онлайн діяльність. Торгівля, яка здійснюється у веб-ресурсі — це продаж продукції або послуг за допомогою мережі Інтернет. В той час, як

мережа Інтернет стала доступною будь-яким користувачам, онлайн-торгівля стала популярною та швидко розповсюджувалася. Перспективи Інтернет торгівлі однозначні в першу чергу в процесі організації. Основним видом електронної торгівлі є спеціалізований сайт, який і буде виступати своєрідним торгівельним майданчиком. Якщо для звичайної системи торгівлі необхідно орендувати магазин, наймати продавців, що коштує дуже багато матеріалу, то для електронної комерції необхідно лише сайт та підтримка його. Характер цих ресурсів – це особлива структура покупки товарів та варіантів оплати, які дозволяють кінцевим користувачам використовувати такі сайти з мінімальними технічними навичками через Інтернет.

На даний час онлайн торгівля активно розвивається, і при бажанні в мережі Інтернет можна придбати будь-який товар або замовити будь-яку послугу. Також онлайн-торгівля надає можливість придбати товари в роздріб або оптом, як зручніше самому споживачу. Знайти ринок збуту становиться набагато легше, будь-який користувач Інтернету зможе з легкістю зробити замовлення за допомогою телефону. Деталі продажу Інтернет також полягають у зникненні географічної приналежності до місця продажу продукту. Сайт працює цілодобово і користувач в зручний час для нього може зробити замовлення, заявки фіксуються в автоматичному режимі, що дає ще одну перевагу інтернет-магазинах. Зрозуміло, що саму заявку буду оброблено в робочий час. Сам споживач може зробити вибір даного товару, але не оформляти замовлення, і через деякий час впевнитись, що саме йому необхідно. В звичайних магазинах не можливо вибрати товар та розмірковувати декілька днів, який саме необхідний споживачеві. Ресурси, які необхідні для утримання інтернет-магазину набагато менші ніж для звичайного магазину, що знижує витрати на їх утримання. Дає можливість такої економії система CMS. Вона працює на основі загальної інформаційної бази, яка надає можливість автоматизувати основні бізнес-процеси, а також створювати повні взаємодії та оперативну інформацію між філіалами компанії. За допомогою цього програмного забезпечення онлайн-торгівля є більш ефективним способом

продажу товару, що збільшує обіг і прибутковість компанії. Своєчасність та швидкість виконання замовлень може значно підвищити довіру покупців[2].

1.3 Аналіз аналогів інтернет-магазинів

Перше враження про сайт, як і про людину, робиться за кілька секунд, в основному, за зовнішнім виглядом. Тому створюючи інтернет-магазин важливо не тільки наповнити його якісними товарами за відмінним цінами, але і підібрати таке оформлення, яке, з одного боку, допоможе зацікавити відвідувачів, а з іншого - не буде відволікати від вибору товарів.

За даними дослідження компанії TemplateMonster, відомого розробника шаблонів сайтів, 42% покупців створюють своє уявлення про інтернет-магазин, ґрунтуючись тільки на його оформленні, а 52% – ніколи не повернуться на сайт, який їм не сподобався [8]. Тому грамотний дизайн інтернет-магазину дійсно допомагає продавати. На сприйняття сайту, в першу чергу, впливають такі елементи оформлення:

- кольору;
- шапка сайту;
- оформлення "Вітрини" на головній сторінці;
- фавікон.

Важливо, щоб усі ці елементи поєднувалися як з тематикою сайту, так і між собою. Тоді інтернет-магазин буде виглядати гармонійно і продумано, а значить робити замовлення на ньому буде зручно і приємно. Фоновий рисунок має асоціюватися з тематикою сайту. Кольори здатні викликати емоції та асоціації, це потрібно враховувати при виборі колірного рішення сайту – від кольору кнопок до фону.

Логічно, що цільова аудиторія інтернет-магазину м'ясний виробів - це користувачі середнього віку, а компанія, яка продає товари для новонароджених - молоді мами. Звичайно, завжди бувають винятки, але орієнтуватися все одно потрібно на переваги більшості покупців. З одягом для

немовлят асоціюються ніжні пастельні відтінки, а з м'ясний виробами - червоні кольори. Значить відтінки цих кольорів і потрібно використовувати.

Для того, щоб зробити сайт більш запам'ятовується, унікальним і створити особливу атмосферу, можна використовувати різні текстури для фону. Але тут також потрібно враховувати особливість аудиторії і товару. Не варто ставити на фон шестерінки в інтернет-магазині дитячого одягу, а ось для сайту, який продає годинники, він буде доречним. Один з головних елементів дизайну будь-якого сайту це його шапка. Вона розташована в самому верху сайту. Саме її клієнт бачить першою, коли відриває сайт, до того ж вона, найчастіше, наскрізна, тобто відображається на всіх сторінках. Тому їй приділяється особлива увага.

Часто початківці інтернет-підприємці використовують як шапки сайту просто симпатичну картинку, яка не має ніякого відношення до продається товару. Це може поставити відвідувачів в тупик. Тому зображення для шапки сайту має підходити по тематиці. При цьому воно повинно органічно вписуватися в загальний дизайн інтернет-магазину. Оформлення вітрини інтернет-магазину має підштовхувати до покупки. Відразу після шапки сайту відвідувач звертає увагу на центральну частину головної сторінки. Для цього використовується "Вітрина". Як і в звичайному магазині, на вітрині сайту розміщуються: найпопулярніші товари, акційні пропозиції, новинки і т.д. Те, що зможе зацікавити і переконати потенційного покупця залишитися на сайті і зробити замовлення. Але важливо не перестаратися: перенасичена яскравими зображеннями і великою кількістю акцій головна сторінка може здатися захаращеною, і відвідувачеві сайту буде не просто знайти потрібну йому інформацію і товар. Не варто завантажувати головну сторінку і занадто довгими текстами. Тут повинна бути тільки найважливіша інформація.

Також варто пам'ятати про те, щоб у товарів були якісні фотографії - великого розміру, привабливі, які відображатимуть реальні кольори товару. Розмиті, маленькі, незрозумілі зображення на головній сторінці можуть зіпсувати враження навіть про добре оформлений інтернет-магазин. Фавікон

допоможе користувачеві швидше знайти сайт. Ще один маленький, але важливий елемент web-дизайну, про який часто забувають – це фавікон (Favicon), невелика картинка, яка відображається поруч з назвою сайту в списку відкритих вкладок браузера.

Оскільки його розмір маленький, то він не може містити дрібних деталей, але в теж час повинен передавати тематику сайту. Найчастіше для фавікона використовують логотип компанії або тематичну.

Нижче наведено приклади вдалого і невдалого оформлення сторінки.

Наприклад, на рисунку 1.1 (<https://kovbasnasprava.com.ua/ua/>) кольори обрані в одній гаммі – і для шапки сайту, і, що важливо, для самих прикладів продукції, запропонованої на сайті.

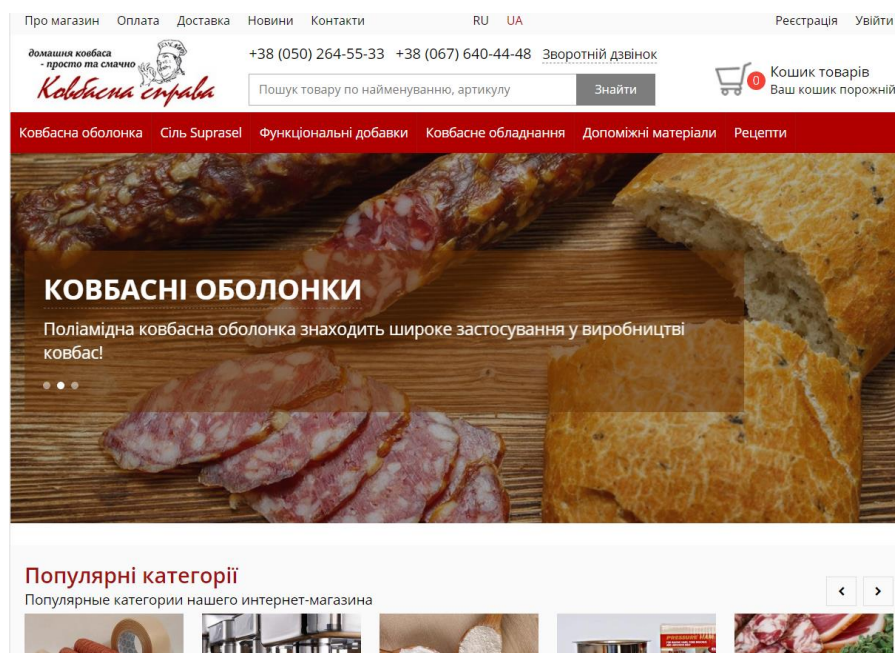


Рисунок 1.1 – Приклад вдалого колірного оформлення
(<https://kovbasnasprava.com.ua/>)

На наступному проаналізованому сайт «Mozzarella» на рисунку 1.2 (<https://mozzarella.com.ua>), окрім невиразного кольору оформлення перенасичена текстовою інформацією фокус уваги майбутнього покупця втрачається але, присутні етнічні мотиви, що можна вважати слідуванням модній тенденції.

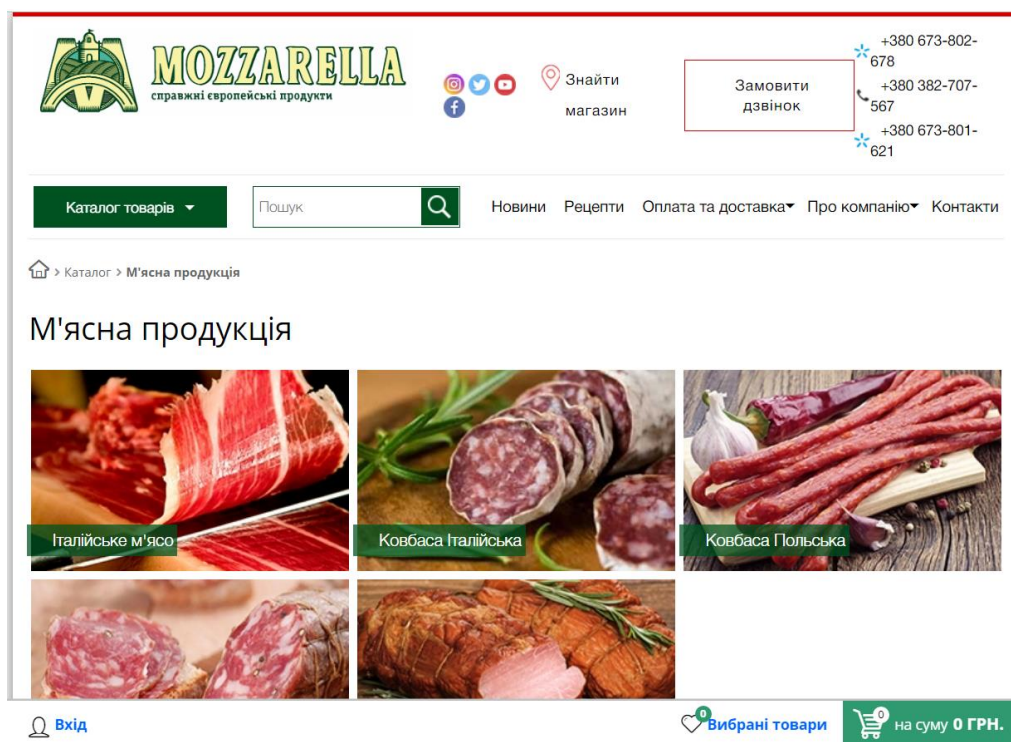


Рисунок 1.2 – Приклад о колірному оформленні вітрини сайту
(<https://mozzarella.com.ua/>)

Колірне оформлення наступної сторінки інтернет-магазину є невдалим – колір фону коричневого кольору, не зовсім асоціюється з м'ясними товарами.

Ще одним суттєвим відштовхуючим елементом є спливаючі вікна з нав'язливою та агресивною рекламою. Варто зазначити, що на багатьох сайтах подібні речі з'являються протягом вже другої-третьої секунди перебування користувача.

Таким чином, можна зробити висновок, що роздрібна торгівля є перспективним напрямком роботи і заробітку за допомогою мережі Інтернет. Але зовнішній вигляд сайту може як привабити потенційних покупців, так і відлякати їх. Тому в процесі розробки необхідно врахувати позитивні моменти в розробки конкурентів і уникнути негативних моментів.

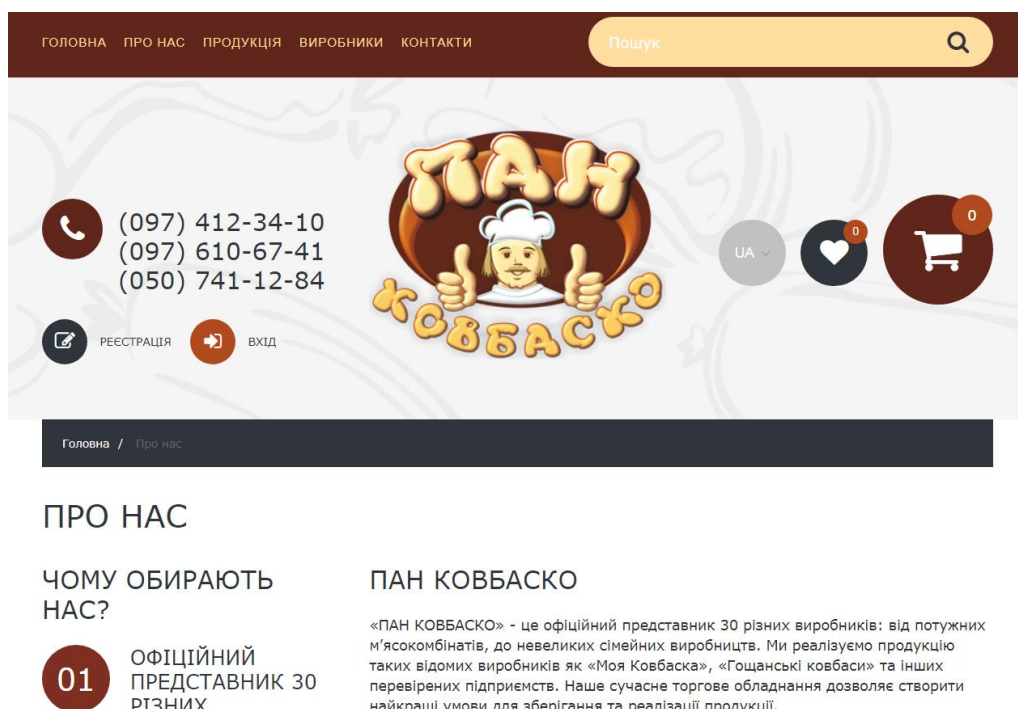


Рисунок 1.3 – Приклад невдалого колірного оформлення
(<https://pankovbasko.com/ru>)

Основною задачею інтернет-магазину є представлення певної, тематичної інформації про товар в Інтернеті, сервісу і підтримки клієнтів, надання онлайн послуг з продажів певної категорії товару. Зручний та логічно побудований інтерфейс дає ряд переваг для звичайного користувача:

- швидкий перехід між сторінками сайту;
- знаходження потрібної інформації;
- мобільність системи.

РОЗДІЛ 2

РОЗРОБЛЕННЯ ВЕБ-РЕСУРСУ

ТА ВИКОРИСТАННЯ МЕТОДІВ МОНІТОРИНГУ В НЬОМУ

2.1 Визначення структури додатку

Основу будь-якої інформаційної системи складає база даних, в якій знадиться вся інформація, що обробляється. Концепції баз даних нараховує дві категорії: «дані» і «модель даних». Поняття «дані» у концепції баз даних - це набір конкретних значень, параметрів, що характеризують об'єкт, умова, ситуацію або будь-які інші фактори [5]. Дані не мають певної структури, дані стають інформацією, коли користувач задає їм певну структуру, тобто смисловий зміст. Тому основним поняттям в області баз даних є поняття моделі даних. Основного терміну моделі даних не існує, у різних авторів ця термін визначається з деякими відмінностями, але, тим не менш, можна виділити загальне означення в цих визначеннях [8].

Основою будь-якої системи баз даних є концептуальна модель. На цьому рівні виконується концептуальне проектування, тобто створення схеми бази даних. Цей етап створює докладні моделі користувацьких уявлень даних, після цього вони інтегруються в концептуальну модель, фіксуючу всі елементи даних, які будуть складовою бази даних.

Предметною областю є складова реальної системи, що забезпечує інтерес для даного дослідження. При проектуванні автоматизованих інформаційних систем предметна область відображається моделями даних, які складаються з декількох рівнів. Число рівнів залежить від складності завдань, але в будь-якому випадку складається з концептуального і логічного рівнів.

Концептуальна модель даних - це модель предметної області. Складова моделі - це об'єкти і взаємозв'язок. Концептуальна модель є засобом спілкування між різними користувачами і тому розробляється без врахування особливостей фізичного представлення даних. При проектуванні концептуальної моделі всі зусилля розробника повинні бути спрямовані в

основному на структурування даних і виявлення взаємозв'язків між ними без розгляду особливостей реалізації і питань ефективності обробки. Проектування концептуальної моделі засноване на основі аналізу вирішуються на цьому підприємстві завдань по обробці даних. Концептуальна модель включає описи об'єктів і їх взаємозв'язків, що представляють інтерес у розглянутій предметній області. Взаємозв'язки між об'єктами є частиною концептуальної моделі і повинні відображатися в базі даних. Взаємозв'язок може охоплювати будь-яке число об'єктів. З іншого боку, кожен об'єкт може брати участь у будь-якому числі зв'язків. Поряд з цим існують взаємозв'язки між атрибутами об'єкта. Розрізняють взаємозв'язку типу: "один до одного", "один до багатьох", "багато до багатьох" [2].

Графічно концептуальну модель зображується у вигляді діаграми «сутність-зв'язок», зображена на рисунку 2.1.

Одна з концептуальних моделей, яка забезпечена конкретною системою керування базами даних, називається логічною моделлю. У випадку який ми застосували база даних створена за допомогою OpenCart і представляє собою реляційну базу даних.

У реляційних моделях даних об'єкти і взаємозв'язки представлені за допомогою таблиць. Кожна таблиця являється одним об'єктом та має склад: рядки і стовпці. Така таблиця в реляційної моделі має назву відношенням.

Схема складається як зі структурної, так і з семантичної інформації. Структурна інформація пов'язана з оголошенням відносин. Семантична інформація виражається безліччю відомих функціональних залежностей між атрибутами відносин, оголошеними в схемі. У зв'язку з цим виникає питання про коректність представленої схеми. Коректною вважається схема, в якій відсутні небажані функціональні залежності між атрибутами [10].

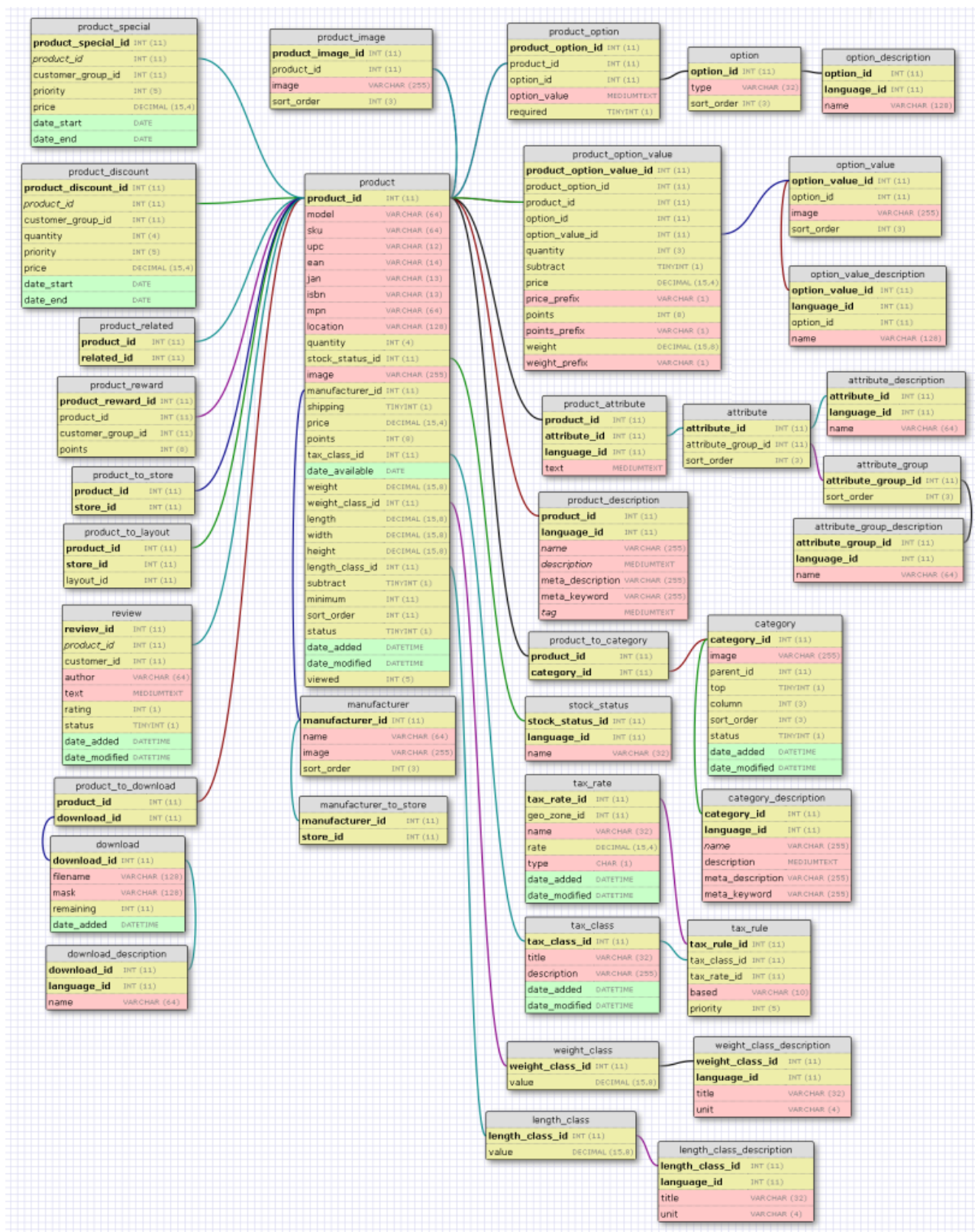


Рисунок 2.1 – Концептуальна модель даних товару

	Телефон	Авторизація, реєстрація	Вкладки	Оформити замовлення
Назва	Пошук		Кошик	
Головне меню (Каталог товару)				
Слайдер				
Інформація	Підтримка	Додаткова інформація	Особистий кабінет	
Підвал				

Рисунок 2.3 – Основний макет сторінок сайту

Взаємозв'язок між розділами й підрозділами сайту (карта сайту) представлено на рисунку 2.4.

Зовнішнє оформлення сторінок виконано за допомогою каскадної таблиці стилів CSS. CSS використовуються для опису зовнішнього вигляду документа, написаного мовою розмітки. Зазвичай CSS-стилі використовуються для створення і зміни стилю елементів веб-сторінок і призначених для користувача інтерфейсів.

Карта сайту

- М'ясо
 - ІНДИЧКА
 - ІНШІ ВИДИ М'ЯСА
 - КУРЯТИНА
 - СВИНИНА
- М'ЯСОВИБАСНИ ВИРОБИ
 - ТМ «СУМСЬКІ КОВБАСИ»
 - ТМ "БАЩИНСЬКИЙ"
- НАПІВФАБРИКАТИ ОХОЛОДЖЕНІ
 - ДЕЛІКАТЕСНА ГРУПА
 - КОВБАСКИ
 - ФАРШІ
 - ШАШЛИК
- НАПІВФАБРИКАТИ ЗАМОРОЖЕНІ
 - ТМ «ЛЕГКО!»
 - ТМ "ПЕЛЬМЕН"
 - ТМ "СИТНІ"
- Акції
- Особистий кабінет
 - Особиста інформація
 - Пароль
 - Мої адреси
 - Історія замовлень
 - Файли для скачування
- Кошик
- Оформлення замовлення
- Пошук
- Інформація
 - ПРО КОМПАНІЮ
 - Доставка
 - Політика конфіденційності
 - Наші контакти

Рисунок 2.4 – Карта сайту

Основним завданням каскадних таблиць стилів описувати правила форматування елементів за допомогою властивостей і допустимих значень цих властивостей. Для кожного елемента можна використовувати обмежений набір властивостей, інші властивості не будуть чинити на нього ніякого впливу.

Оголошення стилю складається з двох частин: елемента веб-сторінки - селектора, і команди форматування - блоку оголошення. Селектор повідомляє браузеру, який саме елемент формувати, а в блоці оголошення (код в фігурних дужках) перераховуються форматує команди - властивості і їх значення.

Селектори представляють структуру веб-сторінки. З їх допомогою створюються правила для форматування елементів веб-сторінки. Селекторами можуть бути елементи, їх класи і ідентифікатори, а також псевдокласи і псевдоелементи.

Для створення таблиці стилю використовувалися такі елементи стилю:

- Float - визначає, по якій стороні буде вирівнюватися елемент, при цьому інші елементи будуть обтікати його з інших сторін,
- Height - встановлює висоту блокових або замінних елементів. Висота не включає товщину меж навколо елемента, значення відступів і полів,
- Width - встановлює ширину блокових або замінних елементів,
- Background - параметр дозволяє встановити одночасно до п'яти атрибутів стилю фону. Значення можуть йти в будь-якому порядку, браузер сам визначить, яке з них відповідає потрібному атрибуту,
- Top - елемент вирівнюється по верху рядки таблиці,
- Left – будь-яких елемент вирівнюють по лівому полі,
- Text-align – вирівнює текстовий елемент,
- Color – задає колір елементу,
- Border - ширина границі блоку,
- Position - встановлює спосіб позиціонування елемента щодо вікна браузера або інших об'єктів на веб-сторінці.

– **Padding** - встановлює значення полів навколо вмісту елементу. Полем називається відстань від внутрішнього краю рамки елемента до уявного прямокутника, що обмежує його вміст

– **Margin** - встановлює величину відступу від кожного краю елемента. Відступом є простір від кордону поточного елемента до внутрішньої межі його батьківського елемента.

В наш час найбільш зручним варіантом для створення та підтримки сайту є створення сайту на базі CMS – системи управління контентом. CMS це система управління сайтом, яка дозволяє додавати інформацію на сайт через спеціальний модуль адміністратора. Тобто не треба створювати окрему HTML-сторінку і кожен раз розташовувати її на хостингу, щоб оновити інформацію або змінити оформлення.

Для розробки даного сайту було обрано CMS ocStore, що додатково дає можливість вести статистику продажу товарів, визначати різні статистичні дані [6].

OcStore(OpenCart) - це нова платформа електронної комерції, є базою для інтернет-торгівлі. Вона забезпечує професійний і надійний фундамент, на основі якого будується інтернет-магазин. OpenCart володіє великою кількістю функцій, що дозволяють здійснювати повний контроль над налаштуваннями магазину [3]. На рис. 2.5 зображено адміністративну панель управління.

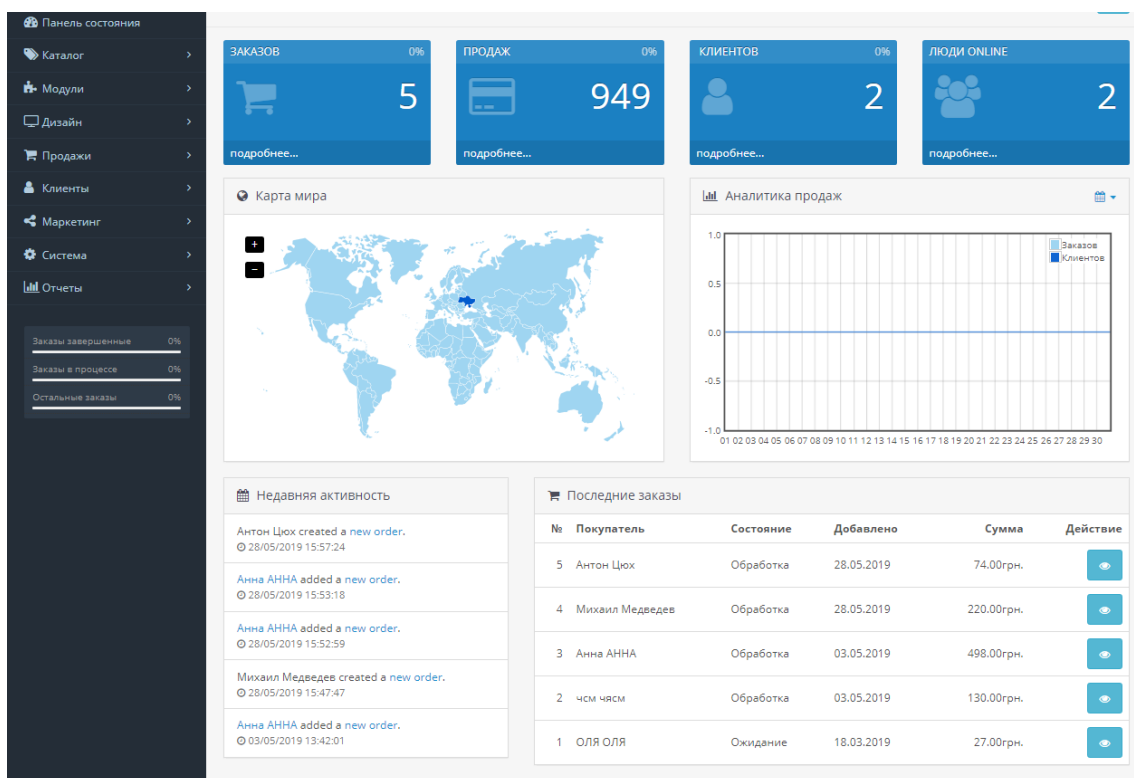


Рисунок 2.5 – Адміністративна сторінка сайту

На рисунку 2.6 показано головну сторінку сайту. На сайті передбачено горизонтальне навігаційне: головне меню, що забезпечує переміщення власне сайтом – сторінки усіма запропонованими категоріями товару, меню в «підвалі» сайту яке забезпечує переміщення не лише за категоріями товару, а і власне сайтом. Подібний дизайн та верстка були обрані для того, щоб не відволікати потенційних покупців від ознайомлення з асортиментом сайту. Причому, кожне з посилань містить вкладені посилання, безпосередньо за різновидами товару.

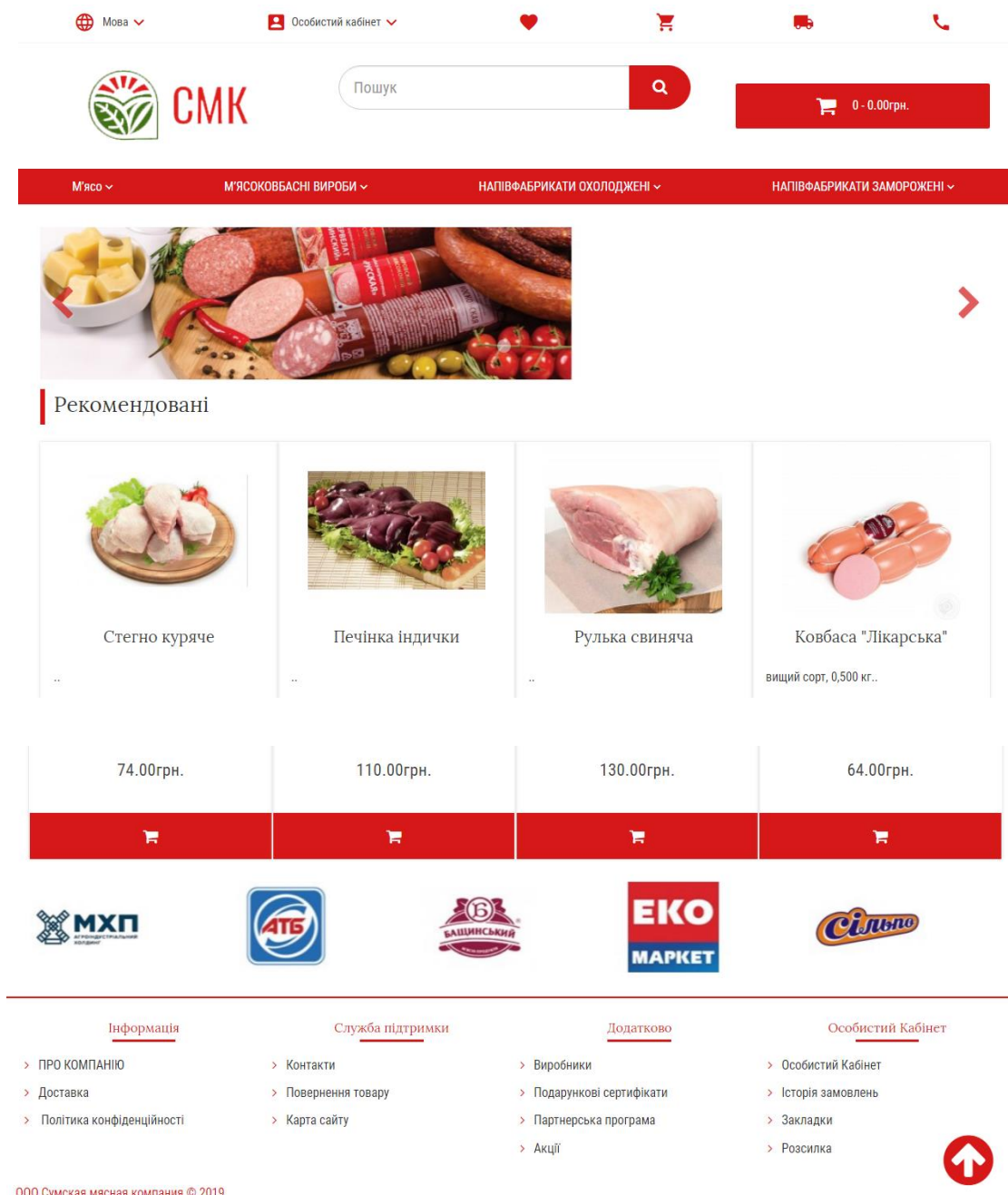


Рисунок 2.6 – Головна сторінка сайту

Головне меню, в першу чергу, призначене для допомоги потенційному покупцеві. За посиланнями головного меню можна отримати інформацію про асортимент продукції, представлений на сайті. Для зручності пошуку і перегляду меню має вигляд категорій косметичних товарів.

Клієнту необхідно пройти реєстрацію, для здійснення покупок. В особистому кабінеті клієнт має багато функцій управління, а саме змінення особистих даних, перегляд історії замовлення, контроль доставки товару[9].

Мова

Особистий кабінет

Реєстрація

Якщо Ви вже зареєстровані, перейдіть на сторінку авторизації.

Основні дані

Ім'я

Ім'я

Прізвище

Прізвище

E-Mail

E-Mail

Телефон

Admin

пароль

Пароль

.....

Підтвердження пароля

Підтвердження пароля

Розсилка новин

Підписка на новини

☐ Так
☒ Ні

Я прочитав [Політика конфіденційності](#) і згоден з умовами

Продовжити

Вхід

Реєстрація

Забули пароль?

Моя інформація

Адресна книга

Закладки

Історія замовлень

Файли для скачування

Регулярні платежі

Бонусні бали

Повернення

Історія транзакцій

E-Mail розсилка

Рисунок 2.7 – Реєстрація користувача

Мова

Особистий кабінет

М'ясо

М'ЯСОКОВБАСНІ ВИРОБИ

НАПІВФАБРИКАТИ ОХОЛОДЖЕНІ

НАПІВФАБРИКАТИ ЗАМОРОЖЕНІ

Особистий Кабінет

Мій обліковий запис

Змінити контактну інформацію

Змінити пароль

Змінити мої адреси

Показати закладки

Мої замовлення

Історія замовлень

Файли для скачування

Запити на повернення

Історія транзакцій

Періодичні платежі

Моя інформація

Змінити контактну інформацію

Пароль

Адресна книга

Закладки

Історія замовлень

Файли для скачування

Регулярні платежі

Бонусні бали

Повернення

Історія транзакцій

E-Mail розсилка

Рисунок 2.8 – Особистий кабінет

Мова
Особистий кабінет

1 - 110.00грн.

М'ясо
М'ЯСОКОВБАСНІ ВИРОБИ
НАПІВФАБРИКАТИ ОХОЛОДЖЕНІ
НАПІВФАБРИКАТИ ЗАМОРОЖЕНІ

Кошик для покупок

Кошик для покупок

Зображення	Назва	Модель	Кількість		Ціна за шт.	Всього
	Печінка індички	Індейка	1		110.00грн.	110.00грн.

Сума: 110.00грн.

Разом: 110.00грн.

Продовжити покупки

Оформлення замовлення

Рисунок 2.9 – Корзина покупок

Мова
Особистий кабінет

1 - 110.00грн.

М'ясо
М'ЯСОКОВБАСНІ ВИРОБИ
НАПІВФАБРИКАТИ ОХОЛОДЖЕНІ
НАПІВФАБРИКАТИ ЗАМОРОЖЕНІ

Кошик для покупок

Оформлення замовлення

Платіжні реквізити

Ім'я

Прізвище

Деталі доставки

Ім'я

Прізвище

Спосіб доставки

Оберіть зручний спосіб доставки для даного замовлення

Доставка по місту

☐ Доставка по місту (Вага: 1.00кг) 11.60грн.

Продам

☒ Самовивіз із магазину 0.00грн.

Спосіб оплати

Виберіть спосіб оплати для даного замовлення

☒ Оплата готівкою

Ви можете додати свій коментар до своїх замовленням

Рисунок 2.10 – Оформлення замовлення

Мова

Особистий кабінет

Кошик для покупок

Оформлення замовлення

Оформлення замовлення

Назва товару	Модель	Кількість	Ціна	Усього
Печінка індички	Індейка	1	110.00грн.	110.00грн.
			Сума:	110.00грн.
			Разом:	110.00грн.

Оплата готівкою. Інструкція

Одержувач платежу:
ООО Сумская мясная компания

Прийом платежів за адресою:
пр. Курский, 147
г. Сумы, Украина, 40030

Ваше замовлення не буде оброблено, поки ми не отримаємо оплату.

Підтвердження замовлення

Назад

Рисунок 2.11 – Історія замовлень

2.3 Розробка функціоналу сайту

Cascading Style Sheets (CSS) - мова таблиць стилів, що використовується для опису презентації документу, написаного мовою HTML або іншою мовою розмітки, наприклад Scalable Vector Graphics (SVG). CSS описує як повинні відображатись структурні елементи на екрані, папері, інших медіа, чи озвучуватись на читачах екрану. Здатність пристосовувати зовнішній вигляд документу в залежності від медійного засобу для виводу інформації є ключовою для CSS[9].

Поряд з типами носіїв в CSS3 включена підтримка різних технічних параметрів пристроїв, на основі яких потрібно завантажувати ті чи інші стилі. Наприклад, можна визначити смартфон з максимальною роздільною здатністю 640 пікселів і для нього встановити одні стильові властивості, а для інших пристроїв інші. Всі характеристики легко комбінуються, тому допустимо поставити стиль тільки для пристроїв альбомного формату з заданим дозволом екрану.

Можливості медіа-запитів не обмежуються виявленням мобільних пристроїв, з їх допомогою можна створювати адаптивний макет. Такий макет підлаштовується під дозвіл монітора і вікна браузера, змінюючи при необхідності ширину макета, число колонок, розміри зображень і тексту. Медіа-запити обмежують ширину макета і при досягненні цього значення (наприклад за рахунок зменшення вікна або при перегляді на пристрої з вказаним розміром) вже застосовується інший стиль.

Коди основних сторінок наведено у додатку А.

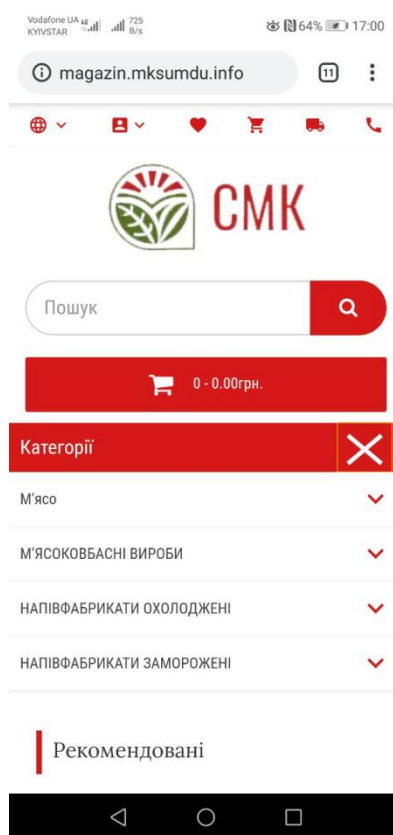


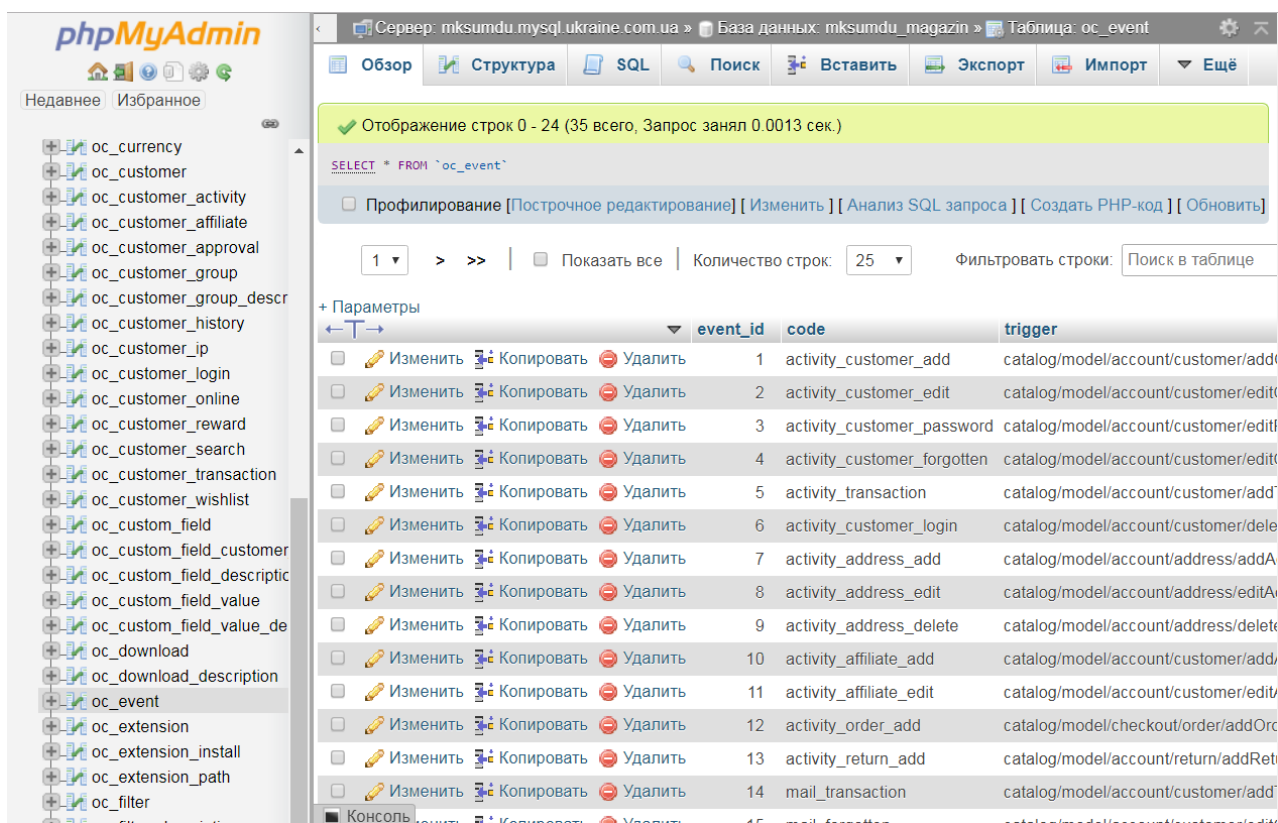
Рисунок 2.12 – Мобільна версія



Рисунок 2.13 – Мобільна версія

Запит – це формулювання своєї інформаційної потреби користувачем деякої бази даних або інформаційної системи, наприклад, пошукової системи. Для складання запиту використовується мова пошукових запитів. Всі запити до пошукових систем умовно (через деяких випадків неоднозначності) можна розділити на три типи.

Команда Select використовується для запитів до бази даних з метою витягання з неї інформації.



Отображение строк 0 - 24 (35 всего, Запрос занял 0.0013 сек.)

SELECT * FROM `oc\_event`

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

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице

	event_id	code	trigger
Изменить Копировать Удалить	1	activity_customer_add	catalog/model/account/customer/add
Изменить Копировать Удалить	2	activity_customer_edit	catalog/model/account/customer/edit
Изменить Копировать Удалить	3	activity_customer_password	catalog/model/account/customer/edit
Изменить Копировать Удалить	4	activity_customer_forgotten	catalog/model/account/customer/edit
Изменить Копировать Удалить	5	activity_transaction	catalog/model/account/customer/add
Изменить Копировать Удалить	6	activity_customer_login	catalog/model/account/customer/dele
Изменить Копировать Удалить	7	activity_address_add	catalog/model/account/address/addA
Изменить Копировать Удалить	8	activity_address_edit	catalog/model/account/address/editA
Изменить Копировать Удалить	9	activity_address_delete	catalog/model/account/address/delete
Изменить Копировать Удалить	10	activity_affiliate_add	catalog/model/account/customer/add
Изменить Копировать Удалить	11	activity_affiliate_edit	catalog/model/account/customer/edit
Изменить Копировать Удалить	12	activity_order_add	catalog/model/checkout/order/addOrc
Изменить Копировать Удалить	13	activity_return_add	catalog/model/account/return/addRet
Изменить Копировать Удалить	14	mail_transaction	catalog/model/account/customer/add
Изменить Копировать Удалить	15	mail_forgotten	catalog/model/account/customer/edit

Рисунок 2.14 – База даних інтернет-магазину

2.4 Тестування інтернет-магазину

Тестування сайту - це перевірка роботи системних процесів на відповідність умов, які забезпечують якість роботи і програмного продукту. Тестування сайту спрямовано на аналіз самої системи, виявленні її недоліків або потенційно уразливих місць згідно системним вимогам [7].

Існує безліч видів тестування, котрі відрізняються як за рівнем аналізу, так і за різними критеріями якості роботи, основні з них:

- Функціональне тестування сайту. Можливість користування інтернет-магазином було розглянуто в пункті 2.3.
- Тестування сумісності. Незалежно від того, з якого гаджета приходять на сайт, інтернет-магазин адаптується. Тестування інтернет-магазину на сумісність було представлено в різних браузерах (рис. 2.15, рис. 2.16, рис. 2.17) та на мобільному додатку (рис 2.18)

– Тестування ефективності. Інтернет-магазин м'ясокомбінату продуктивний, тобто сайт виконує цільові дії в звичайному режимі з урахуванням різних умов.

– Тестування безпеки сайту. При тестуванні безпеки перевіряється – чи немає у користувачів доступу до службових/закритих сторінок, а так само проводиться перевірка захисту всіх критично важливих сторінок (наприклад, розділу адміністрування сайту) від зовнішнього впливу. Якщо при авторизації неправильно ввести пароль, то з'являється повідомлення (рис. 2.18).

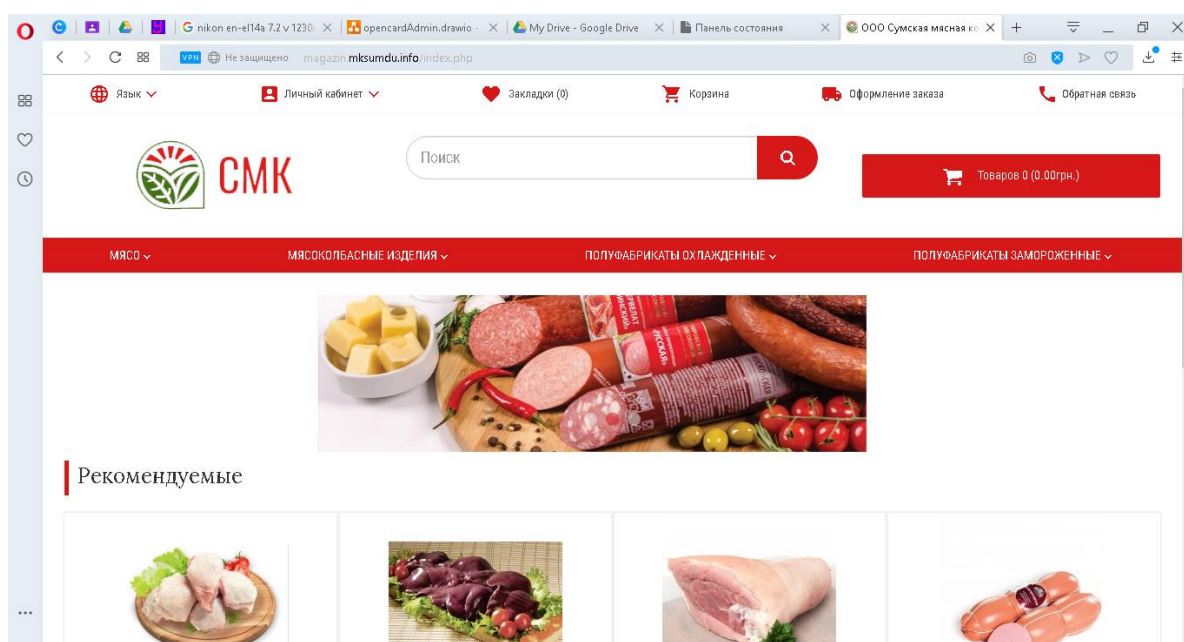


Рисунок 2.15 – Інтернет-магазин в браузері Опера

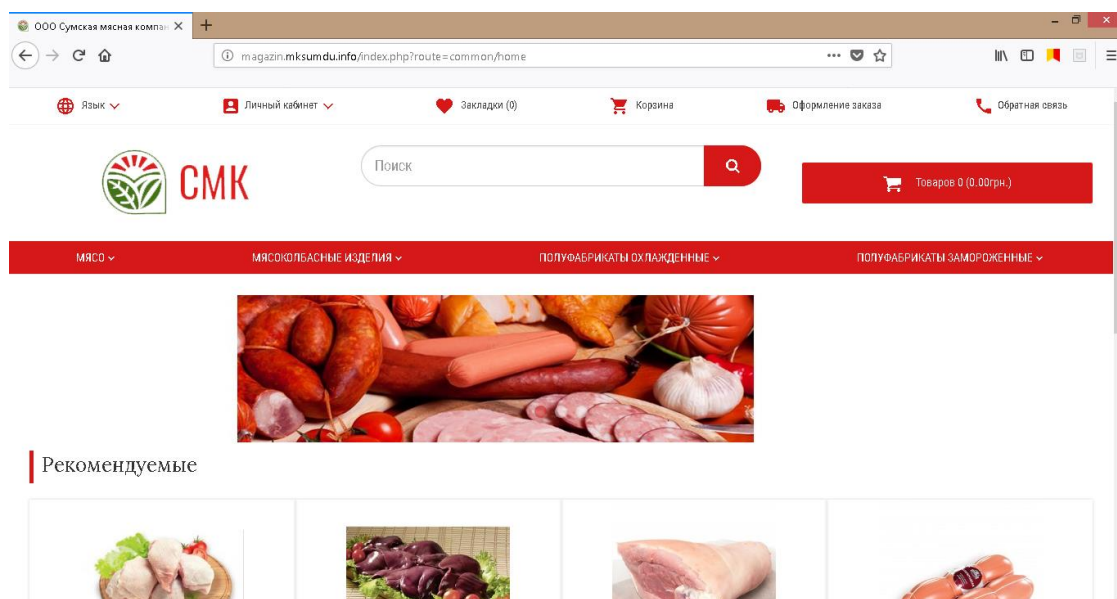


Рисунок 2.16 – Онлайн-додаток в браузері Mozilla Firefox

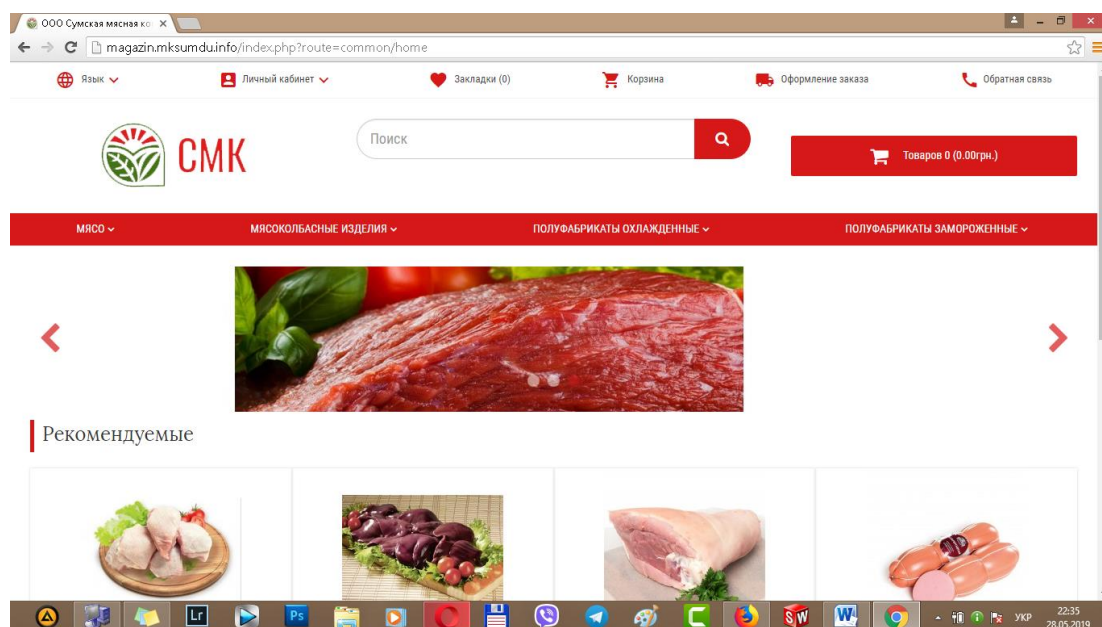


Рисунок 2.17 – Интернет-магазин в браузері Google Chrome

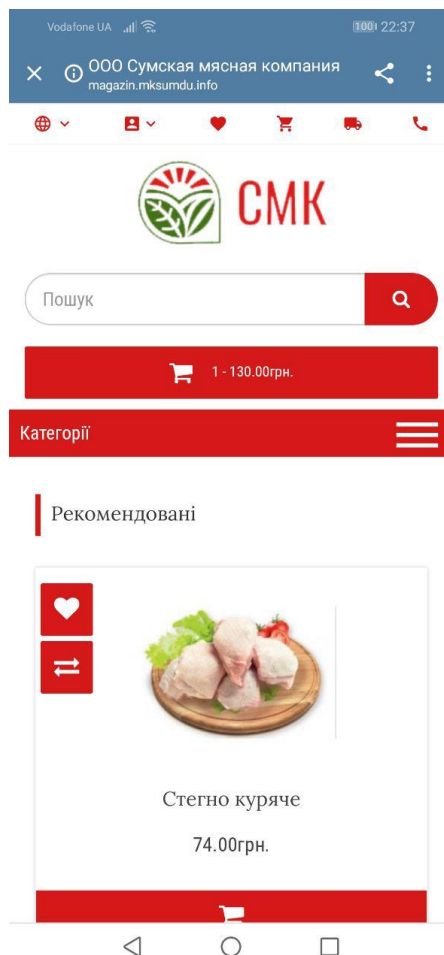


Рисунок 2.18 – Интернет-магазин в мобільній версії

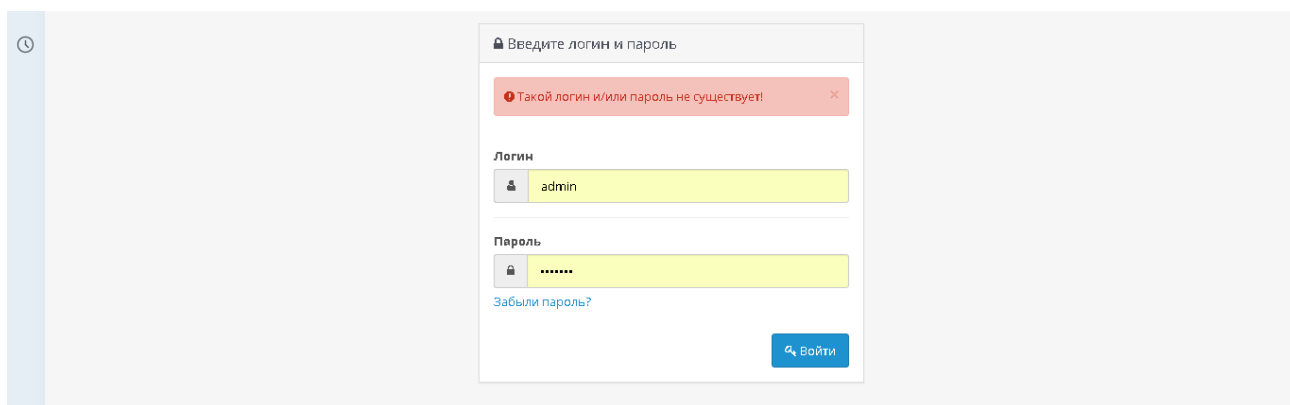


Рисунок 2.19 – Перевірка на правильність логіну та паролю користувача

ВИСНОВКИ

Моніторинг – це система постійного спостереження за явищами і процесами, що проходять в навколишньому середовищі і суспільстві, результати якого служать для обґрунтування управлінських рішень по забезпеченню безпеки людей та об'єктів економіки. На дипломному проектуванні було досліджено та виявлено основні функції моніторингу, а саме:

- Стеження за об'єктом моніторингу. Ключова функція, яка включає в себе періодичний збір - показників з обладнання, сервісів, і т д..
- Зберігання отриманої під час моніторингу інформації. Здійснюється збір інформації за основними показниками і критеріями кожного об'єкта моніторингу. Для зберігання такої інформації використовується бази даних.
- Побудова звітів моніторингу.
- Візуалізація моніторингу. Звіти можуть бути у вигляді текстового подання (логи моніторингу), або у вигляді графіків і діаграм.
- Пошук вузьких місць. На основі аналітичних даних моніторингу представляється можливим дізнатися, яке місце інфраструктури найбільш сильно знижує загальні показники продуктивності всієї системи в цілому.
- Автоматизація сценаріїв. Функція звільняє адміністраторів від рутинних завдань моніторингу.

Завдяки наявності коштів для реалізації всіх вищевикладених функцій адміністратора більше не потрібно перевіряти вручну стан кожної частини системи. Проблеми вирішуються більш оперативно. Діагностика спостерігається системи здійснюється циклічно і точно, а завдяки цьому можна планувати розширення і подальше ускладнення інфраструктури системи за якою спостерігають[5].

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- 1 Моніторинг Інтернету [Електронний ресурс]. – Режим доступу : https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%BD%D1%96%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3_%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82%D1%83
- 2 Online торгівля [Електронний ресурс]. – Режим доступу : <https://soft.ua/articles/online-torgovlya/>
- 3 OpenCart переваги та недоліки [Електронний ресурс]. – Режим доступу : <https://webbuilding.com.ua/ukr/articles/opencart-advantages-disadvantages/>
- 4 Драчов, В.В. Мобільна торгівля як один із перспективних напрямків електронної торгівлі / В.В. Драчов // Журнал європейської економіки. - 2005. - Т. 4, червень. - С. 242-255.
- 5 Електронна комерція в Україні. Статистика за 2015–2016 рр. [Електронний ресурс]. – Режим доступу : <http://nuigde.biz/uk/blog/elektronnaya-kommerciya-v-ukraine-statistika-za-2015-2016-goda.html>
- 6 Ілляшенко С.М. Перспективи та основні проблеми розвитку інтернет-торгівлі в Україні / С.М. Ілляшенко, Т.Є. Іванова // Mechanism of Economic Regulation. – 2014. – № 3. – С. 72–81.
- 7 Методи розробки сайтів [Електронний ресурс]. – Режим доступу: <http://webstudio2u.net>
- 8 Осацька Ю.Є. Міжнародна торгівля: сучасний стан та перспективи розвитку / Ю.Є. Осацька, О.М. Зінченко // Молодий вчений. – 2016. – № 12 (39). – С. 814–816.
- 9 Програмні системи створення веб-сайтів CMS- [Електронний ресурс]. – Режим доступу: <http://www.znannya.org>
8. Юрасов А.Ю. Электронная коммерция: учеб. пособие. – М.: Дело, 2014.

10 Сазонець О.М. Розвиток міжнародної торгівлі за допомогою інформаційних послуг / О.М. Сазонець // Актуальні про-блеми економіки. – 2013. – № 1. – С. 51–57.

Додаток А

Коди основних сторінок

```

{{ header }}
<ul class="breadcrumb">
  {% for breadcrumb in breadcrumbs %}
    <li><a href="{{ breadcrumb.href }}">{{ breadcrumb.text }}</a></li>
  {% endfor %}
</ul>
<div class="content">
  <div class="content-block">
    {{ column_left }}
    <div id="product-search" class="content-center">
      {{ content_top }}
      <h2 class="heading-title">{{ heading_title }}</h2>
      {% if products %}
        <div class="sort-compare-block">
          <div class="compare">
            <div class="form-group"><a href="{{ compare }}" id="compare-total" class="btn btn-link">{{ text_compare }}</a></div>
          </div>
          <div class="sort">
            <div class="form-group input-group input-group-sm">
              <label class="input-group-addon" for="input-sort">{{ text_sort }}</label>
              <select id="input-sort" class="form-control" onchange="location = this.value;">
                {% for sorts in sorts %}
                  {% if sorts.value == '%s-%s'|format(sort, order) %}
                    <option value="{{ sorts.href }}" selected="selected">{{ sorts.text }}</option>
                  {% else %}
                    <option value="{{ sorts.href }}">{{ sorts.text }}</option>
                  {% endif %}
                {% endfor %}
              </select>
            </div>
          </div>
          <div class="sort">
            <div class="form-group input-group input-group-sm">
              <label class="input-group-addon" for="input-limit">{{ text_limit }}</label>
              <select id="input-limit" class="form-control" onchange="location = this.value;">
                {% for limits in limits %}
                  {% if limits.value == limit %}
                    <option value="{{ limits.href }}" selected="selected">{{ limits.text }}</option>

```

```

    {% else %}
    <option value="{{ limits.href }}">{{ limits.text }}</option>
    {% endif %}
  {% endfor %}
</select>
  </div>
</div>
</div>
  <div class="product"> {% for product in products %}
    <div class="product-card">
      <div class="image"><a href="{{ product.href }}"></a></div>
      <div class="caption">
        <h4 class="product-name"><a href="{{ product.href }}">{{
product.name }}</a></h4>
        <p class="description">{{ product.description }}</p>
        {% if product.price %}
        <p class="price"> {% if not product.special %}
          {{ product.price }}
          {% else %} <span class="price-new">{{ product.special
}}</span> <span class="price-old">{{ product.price }}</span> {%
endif %}
          {% if product.tax %} <span class="price-tax">{{ text_tax }}
{{ product.tax }}</span> {% endif %} </p>
          {% endif %}
          {% if product.rating %}
          <div class="rating">
            {% for i in 1..5 %}
              {% if product.rating < i %} <span class="fa fa-stack"><i
class="fa fa-star-o fa-stack-2x"></i></span> {% else %} <span
class="fa fa-stack"><i class="fa fa-star fa-stack-2x"></i><i
class="fa fa-star-o fa-stack-2x"></i></span> {% endif %}
              {% endfor %}
            </div>
          {% endif %} </div>
          <div class="button-top">
            <button type="button" data-toggle="tooltip" title="{{
button_wishlist }}" onclick="wishlist.add('{{ product.product_id
}}');"><i class="fa fa-heart"></i></button>
            <button type="button" data-toggle="tooltip" title="{{
button_compare }}" onclick="compare.add('{{ product.product_id
}}');"><i class="fa fa-exchange"></i></button>
          </div>
          <div class="button-group">
            <button type="button" data-toggle="modal" data-
target="#exampleModalCenter" onclick="cart.add('{{
product.product_id }}', '{{ product.minimum }}');"><i class="fa
fa-shopping-cart"></i> <span class="hidden-xs hidden-sm hidden-
md">{{ button_cart }}</span></button>
          </div>
        </div>
      {% endfor %}
    </div>
  </div>

```

```

    {% endfor %} </div>
<div class="pagination-result-block">
  <div class="result">{{ results }}</div>
  <div class="pagination-block">{{ pagination }}</div>
</div>
{% else %}
<p>{{ text_empty }}</p>
<div class="buttons">
  <div class="pull-right"><a href="{{ continue }}" class="btn
btn-primary">{{ button_continue }}</a></div>
</div>
{% endif %}
</div>
{{ column_right }}</div>
{{ content_bottom }}
</div>
{{ footer }}
{{ header }}
<ul class="breadcrumb">
  {% for breadcrumb in breadcrumbs %}
  <li><a href="{{ breadcrumb.href }}">{{ breadcrumb.text
}}</a></li>
  {% endfor %}
</ul>
<div class="content">
  <div class="content-block">
    {{ column_left }}
    <div id="product-search" class="content-center">
      {{ content_top }}
      <h1 class="heading-title">{{ heading_title }}</h1>
      <label class="control-label" for="input-search">{{
entry_search }}</label>
      <div class="search-wrapp">
        <div class="coll">
          <input type="text" name="search" value="{{ search }}"
placeholder="{{ text_keyword }}" id="input-search" class="form-
control" />
          <label class="checkbox-inline">
            {% if description %}
            <input type="checkbox" name="description" value="1"
id="description" checked="checked" />
            {% else %}
            <input type="checkbox" name="description" value="1"
id="description" />
            {% endif %}
            {{ entry_description }}</label>
          </div>
          <div class="coll">
            <select name="category_id" class="form-control">
<option value="0">{{ text_category }}</option>
            {% for category_1 in categories %}
            {% if category_1.category_id == category_id %}

```



```

    <div class="sort">
        <div class="form-group input-group input-group-sm">
            <label class="input-group-addon" for="input-sort">{{ text_sort
            }}</label>
            <select id="input-sort" class="form-control" onchange="location
            = this.value;">
                {% for sorts in sorts %}
                {% if sorts.value == '%s-%s'|format(sort, order) %}
                <option value="{{ sorts.href }}" selected="selected">{{
                sorts.text }}</option>
                {% else %}
                <option value="{{ sorts.href }}">{{ sorts.text }}</option>
                {% endif %}
                {% endfor %}
            </select>
        </div>
    </div>
    <div class="sort">
        <div class="form-group input-group input-group-sm">
            <label class="input-group-addon" for="input-limit">{{ text_limit
            }}</label>
            <select id="input-limit" class="form-control" onchange="location
            = this.value;">
                {% for limits in limits %}
                {% if limits.value == limit %}
                <option value="{{ limits.href }}" selected="selected">{{
                limits.text }}</option>
                {% else %}
                <option value="{{ limits.href }}">{{ limits.text }}</option>
                {% endif %}
                {% endfor %}
            </select>
        </div>
    </div>
</div>
<div class="product">
    {% for product in products %}
    <div class="product-card">
        <div class="image"><a href="{{ product.href }}"></a></div>
        <div class="caption">
            <h4 class="product-name"><a href="{{ product.href }}">{{
            product.name }}</a></h4>
            <p class="description">{{ product.description }}</p>
            {% if product.price %}
            <p class="price">
                {% if not product.special %}
                {{ product.price }}
                {% else %}
                <span class="price-new">{{ product.special }}</span> <span
                class="price-old">{{ product.price }}</span>
            </p>
            </div>
        </div>
    </div>
    </div>

```

```

        {% endif %}
        {% if product.tax %}
        <span class="price-tax">{{ text_tax }} {{ product.tax
    }}</span>
        {% endif %}
    </p>
    {% endif %}
    {% if product.rating %}
    <div class="rating">
        {% for i in 1..5 %}
        {% if product.rating < i %}
        <span class="fa fa-stack"><i class="fa fa-star-o fa-stack-
2x"></i></span>
        {% else %}
        <span class="fa fa-stack"><i class="fa fa-star fa-stack-
2x"></i><i class="fa fa-star-o fa-stack-2x"></i></span>
        {% endif %}
        {% endfor %}
    </div> {% endif %}
    </div> <div class="button-top">
        <button type="button" data-toggle="tooltip" title="{{
button_wishlist }}" onclick="wishlist.add('{{ product.product_id
}}');"><i class="fa fa-heart"></i></button>
        <button type="button" data-toggle="tooltip" title="{{
button_compare }}" onclick="compare.add('{{ product.product_id
}}');"><i class="fa fa-exchange"></i></button>
    </div>
    <div class="button-group">
        <button type="button" data-toggle="modal" data-
target="#exampleModalCenter" onclick="cart.add('{{
product.product_id }}', '{{ product.minimum }}');"><i class="fa
fa-shopping-cart"></i> <span class="hidden-xs hidden-sm hidden-
md">{{ button_cart }}</span></button>
    </div>
</div>
    {% endfor %}
</div>
    <div class="pagination-result-block">
        <div class="result">{{ results }}</div>
        <div class="pagination-block">{{ pagination }}</div>
    </div>
    {% else %}
    <p>{{ text_empty }}</p>
    {% endif %}
</div>
    {{ column_right }}</div>
    {{ content_bottom }}
</div>
<script type="text/javascript"><!--
$('#button-search').bind('click', function() {
url = 'index.php?route=product/search';
var search = $('.content input[name=\'search\']').prop('value');

```

```

if (search) {
url += '&search=' + encodeURIComponent(search);
}
var          category_id          =          $('#content
select[name=\'category_id\']').prop('value');
if (category_id > 0) {
url += '&category_id=' + encodeURIComponent(category_id);
}
var          sub_category          =          $('#content
input[name=\'sub_category\']:checked').prop('value');
if (sub_category) {
url += '&sub_category=true';
}
var          filter_description    =          $('#content
input[name=\'description\']:checked').prop('value');
if (filter_description) {
url += '&description=true';
}
location = url;
});
$('#content input[name=\'search\']').bind('keydown', function(e)
{
if (e.keyCode == 13) {
$('#button-search').trigger('click');
}});
$('#select[name=\'category_id\']').on('change', function() {
if (this.value == '0') {
$('#input[name=\'sub_category\']').prop('disabled', true);
else {
          $('#input[name=\'sub_category\']').prop('disabled',
false);
        }
    });
$('#select[name=\'category_id\']').trigger('change');
--></script>
{{ footer }}

```