

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Розробка веборієнтованої системи для вибору та
замовлення напоїв»**

Виконав студент групи ПЗ-20-1
КОЛОМІЄЦЬ Артем Олександрович

Керівник роботи:
ФАНТ Микола Олександрович

Рецензент:
ВЛАСЕНКО Олег Васильович

Житомир – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення
Тетяна ВАКАЛЮК
«26» лютого 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **КОЛОМІЄЦЬ Артем Олександрович**

Керівник роботи: **ФАНТ Микола Олександрович**

Тема роботи: **«Розробка веборієнтованої системи для вибору та замовлення напоїв»,**

затверджена Наказом закладу вищої освіти від **«23» лютого 2024 р., №74/с**

Вихідні дані для роботи: **об'єктом дослідження є процеси для вибору та замовлення напоїв, а предметом дослідження є засоби, алгоритми та методи, що забезпечують ефективну роботу системи під час вибору та замовлення напоїв.**

Консультанти з кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Фант М.О.	26.02.2024р.	26.02.2024р.
2	Фант М.О.	11.03.2024р.	11.03.2024р.
3	Фант М.О.	24.04.2024р.	24.04.2024р.

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з вебсайту та пояснювальної записки. Пояснювальна записка до кваліфікаційної роботи містить текстову частину, викладену на 56 сторінках друкованого тексту. Список використаних джерел містить 13 найменувань і займає 1 сторінку. У роботі наведено 38 рисунків. Також у роботі є 15 сторінок додатків. Загальний обсяг роботи складає 72 сторінок.

Метою роботи є розробка веборієнтованої системи, яка забезпечить ефективні вибір та замовлення напоїв.

У результаті огляду й аналізу можливих архітектурних рішень для системи було обрано архітектуру MVC.

Для розробки системи були використані сучасні вебтехнології – PHP, JavaScript, jQuery та MySQL.

Система пропонує сучасний, зручний та адаптивний, інтерфейс, який забезпечує інтуїтивний процес вибору та замовлення напоїв.

Система пройшла необхідні види тестування.

КЛЮЧОВІ СЛОВА: ВЕБСАЙТ, JAVASCRIPT, MVC, MYSQL, PHP.

					ІПЗ.КР.Б-121-24-ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Коломієць А.О.			Розробка веборієнтованої системи для вибору та замовлення напоїв. Пояснювальна записка	Літ.	Арк.	Аркушів
Керівник		Фант М.О.					3	72
						Житомирська політехніка, група ІПЗ-20-1		
Н. контр.		Тичина О.П.						
Зав.каф.		Вакалюк Т.А.						

ABSTRACT

The bachelor's thesis consists of a website and an explanatory note. The explanatory note to the qualification paper contains a text part laid out on 56 pages of printed text. The list of used sources contains 13 names and occupies 1 page. The work contains 38 drawings. There are also 15 pages of appendices in the work. The total volume of work is 72 pages.

The goal of the work is the development of a web-oriented system that will ensure efficient selection and ordering of drinks.

As a result of the review and analysis of possible architectural solutions for the system, the MVC architecture was chosen.

Modern web technologies - PHP, JavaScript, jQuery and MySQL - were used to develop the system.

The system offers a modern, convenient and adaptive interface that provides an intuitive process of choosing and ordering drinks.

The system has passed the necessary types of testing.

KEYWORDS: WEBSITE, JAVASCRIPT, MVC, MYSQL, PHP

.

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ НАПРЯМІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ СИСТЕМИ	9
1.1 Постановка задачі.....	9
1.2 Аналіз аналогів програмного продукту	11
1.3 Вибір архітектури веборієнтованої системи	15
1.4 Вибір й обґрунтування інструментальних засобів для реалізації системи.....	19
Висновки до першого розділу.....	22
РОЗДІЛ 2 ПРОЄКТУВАННЯ СИСТЕМИ	23
2.1 Визначення варіантів використання системи	23
2.2 Розробка бази даних	25
2.3 Розробка функціональних алгоритмів роботи програми.....	27
2.4 Реалізація системи	30
Висновки до другого розділу.....	39
РОЗДІЛ 3 ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБОРІЄНТОВАНОЮ СИСТЕМОЮ	40
3.1 Структура інтерфейсу системи.....	40
3.2 Тестування системи	45
Висновки до третього розділу	53
ВИСНОВКИ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	55
ДОДАТКИ.....	56

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних

ООП – об’єктно-орієнтоване програмування

ПЗ – програмне забезпечення

AJAX – асинхронний JavaScript та XML – підхід до побудови вебсторінок за якого у фоновому режимі відправляється запит на сервер та отримує деякі дані у відповідь

CMS – система управління контентом

CRUD – основні операції, які використовуються для управління даними (створення, читання, оновлення, видалення)

MVC – архітектурний шаблон, який використовується під час проєктування та розробки програмного забезпечення, що дозволяє розділити додаток на три компоненти: Model, View, Controller

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми пов'язана із тим, що у сучасному світі вебсайти є невід'ємною частиною нашого повсякденного життя і їх актуальність щодня тільки зростає. Вони надають можливість людям отримати структуровану та сучасну інформацію на бажану тему, придбати будь-що онлайн, підтримувати зв'язок з рідними та близькими і просто розбавити своє дозвілля. Все це можна робити з будь-якого пристрою: чи то комп'ютер, чи то телефон, чи то планшет.

Неймовірно популярними у користувачів сьогодні стали різні системи для замовлення товарів онлайн. Це зрозуміло, адже відвідування онлайн-магазину не потребує додаткового часу, а великий асортимент і зручність пошуку товарів вражають. На більшості сайтів є зручні фільтри, тому отримати результати пошуку, які задовольняють потреби замовника дуже просто.

Сервіс для замовлення товарів онлайн, якщо він правильно та якісно організований, може давати власнику дуже гарний прибуток та можливість реалізовувати продукцію за межами свого міста чи регіону і навіть по всій країні, а то й світі.

Метою роботи є розробка веборієнтованої системи, яка забезпечить ефективні вибір та замовлення напоїв.

Реалізація поставленої мети потребує виконання таких **завдань**:

1. Виконати огляд й аналіз аналогів системи, дослідити існуючі рішення, їхні переваги та недоліки.
2. Здійснити вибір із обґрунтуванням методів і засобів побудови та проєктування.
3. Розробити систему з використанням вибраних інструментів, спроектувати архітектуру та створити основні компоненти системи.
4. Провести різного виду тестування системи.

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

Об’єктом дослідження даної кваліфікаційної роботи бакалавра є процеси для вибору та замовлення напоїв.

Предметом дослідження є засоби, алгоритми та методи, що забезпечують ефективну роботу системи під час вибору та замовлення напоїв.

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ НАПРЯМІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ СИСТЕМИ

1.1 Постановка задачі

На сьогоднішній день існує велика різноманітність алкогольних напоїв, які користуються популярністю серед споживачів. Їхнє помірне вживання допомагає людям зняти стрес, розслабитися та насолодитися відпочинком. З розвитком технологій дедалі більшої популярності набуває замовлення алкогольних напоїв онлайн.

Головною метою кваліфікаційної роботи є створення вебсайту для вибору та замовлення таких продуктів. Система має бути оснащеною приємним та інтуїтивно зрозумілим інтерфейсом, а також мати функціонал, притаманний подібним вебдодаткам. Користувачі повинні мати можливість ознайомитися з асортиментом напоїв, замовити бажаний товар, створити власний обліковий запис та скористатися зворотнім зв'язком у разі потреби.

На головній сторінці сайту будуть представлені алкогольні напої, доступні для замовлення, що дозволить користувачам швидко переглянути актуальні пропозиції. Картка товару включатиме інформацію про напій, його характеристики, а також галерею зображень для детального ознайомлення. Користувачі зможуть додавати обрані товари до кошика та зберігати їх до моменту оформлення замовлення. Процес оформлення замовлення буде спрощеним для забезпечення максимальної зручності використання системи.

Для адміністраторів сайту буде доступна система управління контентом (CMS), яка дозволить додавати нові товари, редагувати інформацію про наявні продукти та видаляти неактуальні позиції. Система також забезпечуватиме можливість управління профілями користувачів, включаючи налаштування ролей та доступів, а також управління індивідуальними налаштуваннями користувачів.

CMS — це програмне забезпечення, яке дозволяє користувачам створювати, редагувати, управляти та публікувати цифровий контент без

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

необхідності мати глибокі знання у програмуванні. Основною метою CMS є спрощення процесу створення і керування вебсайтами та їх вмістом, що дозволяє зосередитися на самій інформації та її якості.

Основні характеристики CMS:

1. Інтерфейс користувача: зазвичай CMS мають зручний і зрозумілий графічний інтерфейс, який дозволяє користувачам легко працювати з контентом.

2. Управління контентом: CMS забезпечує функції для створення, редагування та організації контенту, включаючи текст, зображення, відео та інші медіафайли.

3. Публікація контенту: системи управління контентом дозволяють легко публікувати контент на вебсайті, контролюючи, як і коли цей контент буде доступний відвідувачам.

4. Управління користувачами та ролями: CMS зазвичай підтримують функціонал для управління користувачами, включаючи призначення ролей та прав доступу. Це дозволяє обмежити або розширити доступ до різних частин системи для різних користувачів.

5. Шаблони та теми: CMS часто пропонують шаблони та теми для оформлення вебсайтів, що дозволяє швидко змінювати зовнішній вигляд сайту без необхідності переписувати код.

6. Модулі та плагіни: більшість сучасних CMS підтримують модулі або плагіни, які додають новий функціонал, що дозволяє розширити можливості базової системи.

SEO та маркетингові інструменти: Вбудовані інструменти для оптимізації сайту для пошукових систем (SEO) та маркетингових кампаній допомагають покращити видимість сайту в інтернеті.

Переваги використання CMS:

1. Простота використання: дружній інтерфейс дозволяє навіть недосвідченим користувачам легко керувати контентом.

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Ефективність: швидке додавання та редагування контенту зменшує час на підтримку та оновлення сайту.

3. Гнучкість: можливість легко розширювати функціонал сайту за допомогою модулів або плагінів.

4. Безпека: регулярні оновлення та патчі забезпечують захист від відомих вразливостей.

CMS-системи широко використовуються для створення та управління різними типами вебсайтів, від особистих блогів до великих корпоративних порталів і інтернет-магазинів.

Загалом, робота включає розробку CMS-системи для адміністрування вебсайту з вибору та замовлення алкогольних напоїв. Етапами розробки є аналіз теоретичних засад проєктування та реалізації CMS, аналіз існуючих вебсайтів за схожою тематикою та визначення найбільш актуального напрямку розробки. Розробка адаптивного інтерфейсу для вебсайту здійснюватиметься за допомогою HTML, CSS та JavaScript, з використанням фреймворку Bootstrap.

Серверна частина буде реалізована на мові PHP у середовищі програмування PhpStorm, а для зберігання даних про користувачів, продукти та замовлення буде використано БД MySQL. Це включає розробку математичної та алгоритмічної моделі функціонування системи на основі бази даних, проєктування бази даних для зберігання інформації про користувачів, продукти та замовлення, а також написання серверної частини для реалізації необхідних функцій CMS-системи: додавання, редагування та видалення продуктів, аутентифікація та управління користувачами, а також обробка замовлень та зміна їх статусу.

1.2 Аналіз аналогів програмного продукту

На просторах Інтернету існує велика кількість різних аналогів за даною тематикою. Розглянемо декілька з них.

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		11

Одним із аналогів є сайт goodwine.com.ua [1]. Головна сторінка цього сайту представлена на рисунку (рис 1.1). Він має доволі приємний дизайн, пропонує різні види алкогольної продукції та, додатково, товари, пов'язані з нею.

Переваги:

- широкий вибір алкогольних напоїв та супутніх товарів;
- великий спектр фільтрів та сортувань.

Недоліки:

- відсутність прямого зв'язку прямо на сайті;
- мова українського сайту – російська.

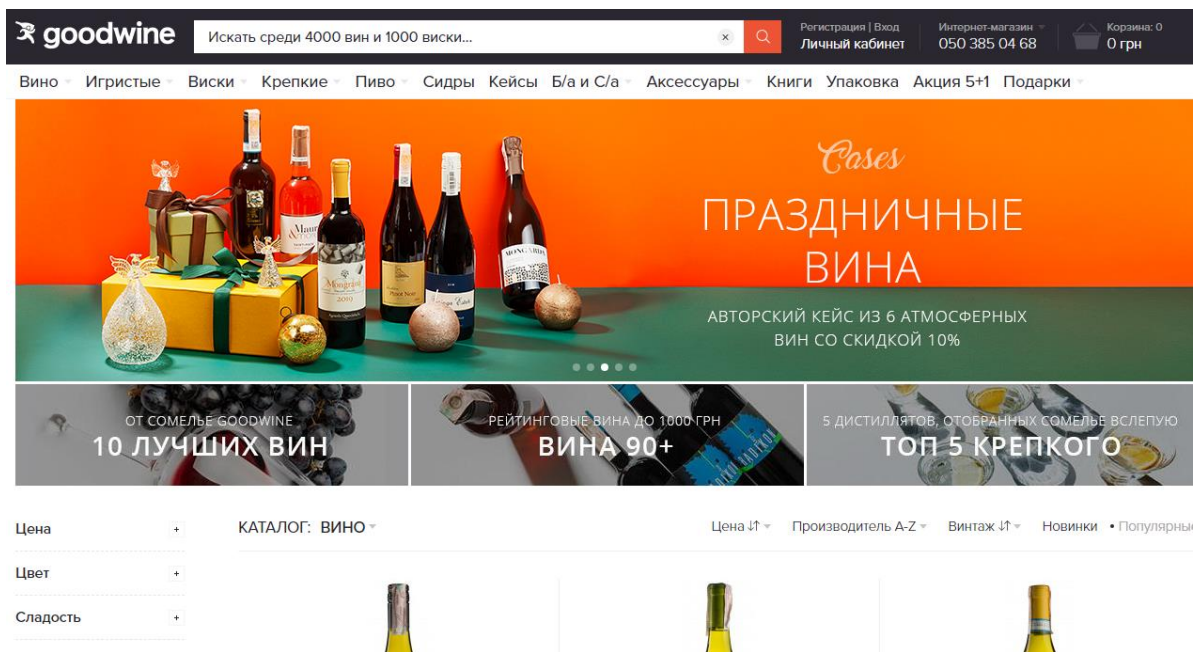


Рисунок 1.1 - Вигляд головної сторінки сайту goodwine.com.ua

Наступним аналогом є сайт vintagemarket.com.ua [2]. На рисунку (рис 1.2) показана головна сторінка вебсайту. У цього сайту більш привабливий та строгий дизайн порівняно з першим. Також він має багато різних видів продукції. В цілому сайт є більш довершеним і наповненим, відносно попереднього.

Переваги:

- насамперед головна особливість, на мій погляд, можливість зміни мови сайту;
- є програма лояльності;

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

- велика кількість фільтрів.

Недоліки:

- ряд проблем пов'язаних з кошиком: якщо кошик пуста – вона вмикається і одразу вимикається. Також в кошику можна обрати мінусову кількість товарів (рис. 1.3);

- відсутність прямого зв'язку прямо на сайті.

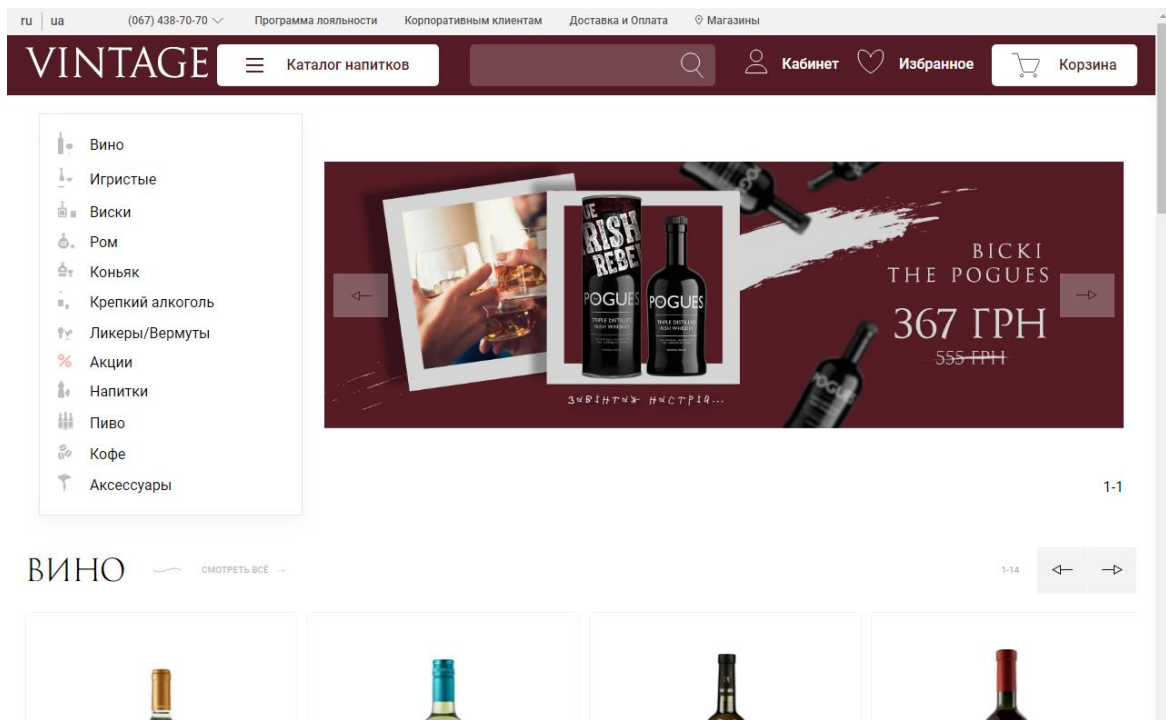


Рисунок 1.2 - Вигляд сайту vintagemarket.com.ua

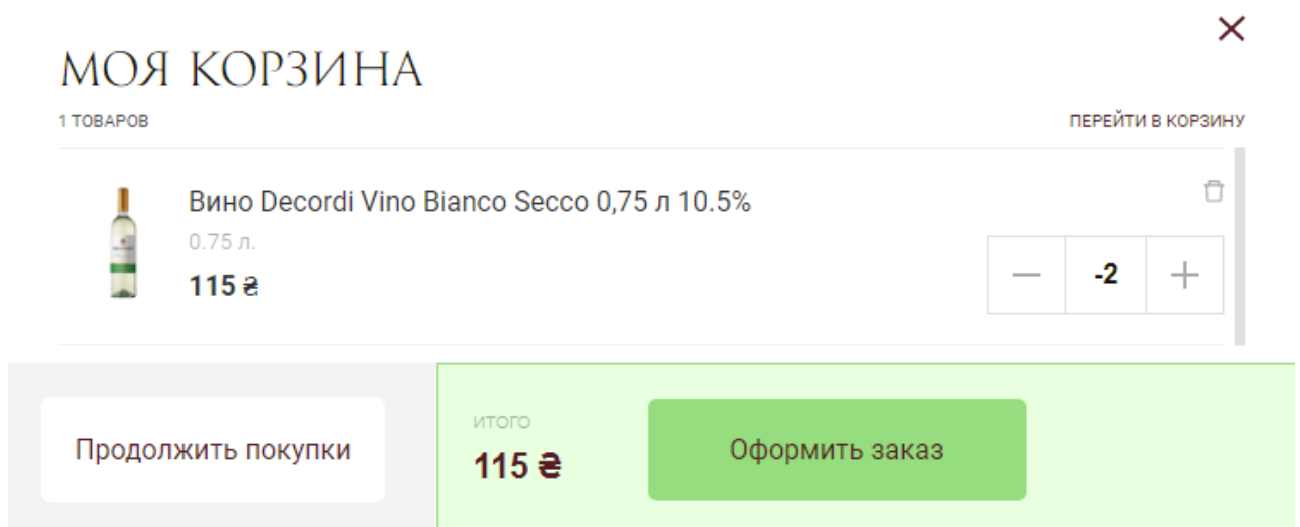


Рисунок 1.3 - Проблема з кошиком на сайті vintagemarket.com.ua

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

Ще одним аналогом є сайт elitochka.net [3]. Порівняно з усіма попередніми сайтами він має найменш привабливий інтерфейс (рис. 1.4). Відповідно користуватися сайтом не дуже приємно та зручно. Проте, слід зауважити, що на рахунок функціоналу, вебдодаток досить наповнений.

Переваги:

- система відгуків клієнтів;
- можливість зміни мови сайту (проте працює недосконало).

Недоліки:

- головним недоліком сайту є відсутність можливості зареєструватися на ньому та створити власний акаунт;
- відсутність фільтрації за ціною;
- непривабливий дизайн.

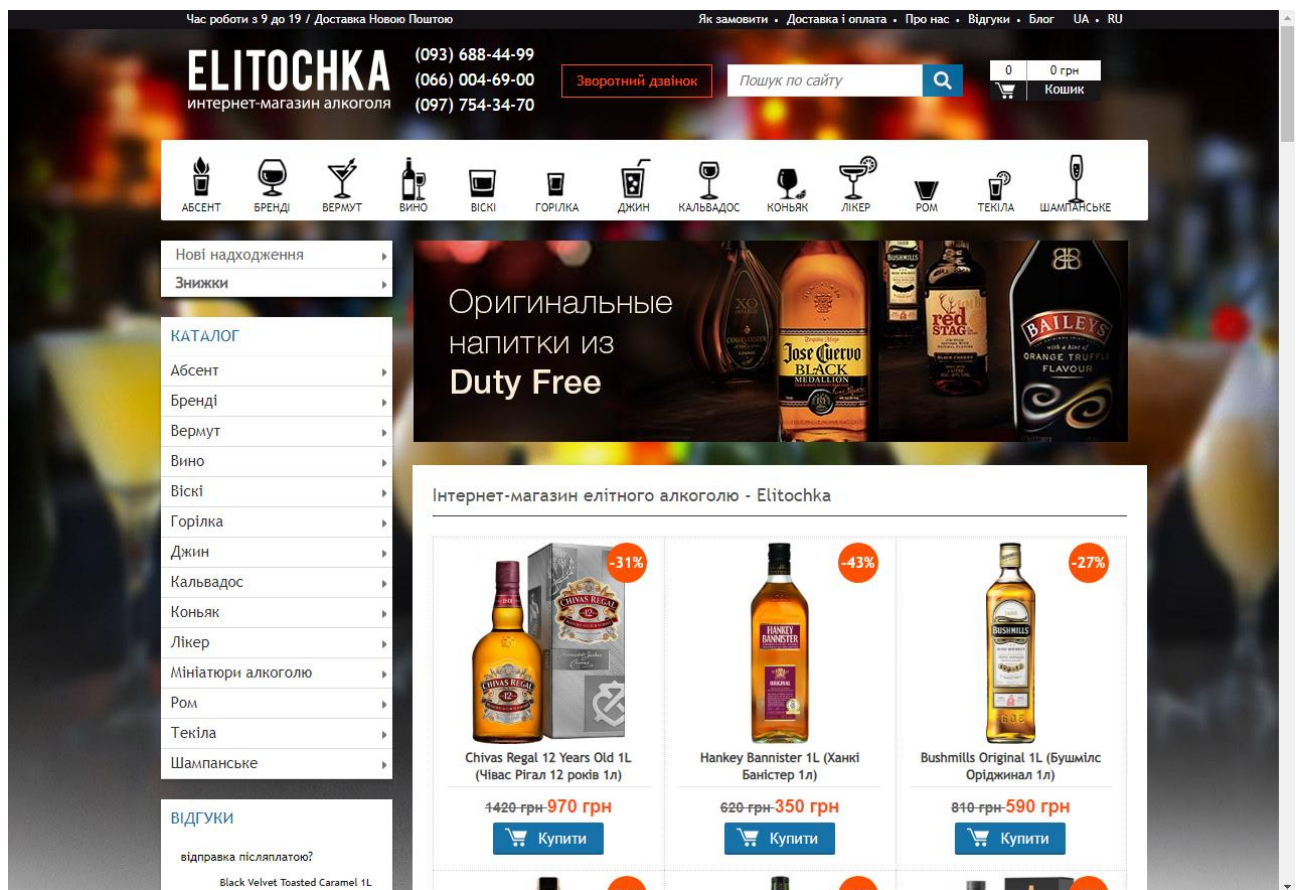


Рисунок 1.4 - Вигляд сайту elitochka.net

У результаті порівняння аналогів були визначені вимоги до розроблюваної системи: зручний та привабливий інтерфейс, українська мова інтерфейсу, зручна система для сортування та фільтрації напоїв.

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

1.3 Вибір архітектури веборієнтованої системи

У даній роботі потрібно розробити вебсайт, який надаватиме можливості реєстрації, авторизації, пошуку напоїв, отримання детальної інформації про них, вибору та замовлення напоїв і можливості зворотного зв'язку.

При розробці даного вебсайту для проєктування ПЗ використовуватиметься об'єктно-орієнтований підхід, адже він має безліч переваг над іншими методами програмування.

Об'єктно-орієнтоване програмування (ООП) — це парадигма програмування, яка базується на концепції "об'єктів", які можуть містити дані у вигляді полів (часто відомих як атрибути або властивості), і коду у вигляді методів (функцій). Основна ідея ООП полягає в об'єднанні даних і методів, що працюють з цими даними, в одну структуру, яка називається об'єктом. Це дозволяє створювати більш структуровані, зрозумілі та підтримувані програми [4].

Інкапсуляція є однією з основних засад ООП. Вона передбачає об'єднання даних і методів, що працюють з цими даними, в єдиний об'єкт. Інкапсуляція дозволяє приховувати внутрішню реалізацію об'єкта від зовнішнього світу і забезпечує контрольований доступ до даних через публічні методи. Це сприяє безпеці даних і зменшує залежність між різними частинами програми, що робить її більш гнучкою та легко підтримуваною.

Наслідування є ще однією ключовою концепцією ООП. Воно дозволяє створювати нові класи на основі існуючих, успадковуючи їх властивості та методи. Це сприяє повторному використанню коду і зменшує дублювання. Наприклад, можна створити базовий клас "Транспортний засіб" з загальними властивостями, такими як швидкість і колір, і потім створити класи "Автомобіль" і "Мотоцикл", які успадковують ці властивості і додають свої унікальні характеристики [5].

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

Поліморфізм дозволяє об'єктам різних класів бути обробленими за допомогою однакових інтерфейсів. Це означає, що один метод може мати різну реалізацію залежно від об'єкта, який його викликає. Поліморфізм сприяє створенню гнучкого та розширюваного коду. Наприклад, метод "малювати" може бути реалізований по-різному для об'єктів класів "Круг", "Квадрат" і "Трикутник", але всі вони можуть бути оброблені через спільний інтерфейс "Фігура".

Абстракція дозволяє виділяти лише найважливіші характеристики об'єкта, приховуючи всі деталі реалізації, які не є суттєвими для користувача. Це допомагає зменшити складність системи та зробити її більш зрозумілою. Абстракція дозволяє програмістам працювати з високорівневими концепціями, не заглиблюючись у деталі низького рівня. Наприклад, при роботі з класом "Банк", користувач може бачити методи для створення рахунку, внесення коштів і зняття грошей, але не знає, як ці операції реалізовані всередині.

ООП пропонує багато переваг для розробників програмного забезпечення. Основною перевагою є можливість створення модульного, повторно використовованого та масштабованого коду. Завдяки ООП можна легко оновлювати та розширювати існуючі програми, додавати новий функціонал без значних змін в основній структурі. Це робить програми більш підтримуваними та знижує вартість їхнього обслуговування.

Модульність є важливою перевагою ООП, оскільки дозволяє розділити програму на незалежні компоненти, які можна легко тестувати, оновлювати та повторно використовувати. Це сприяє зниженню вартості розробки та підвищенню якості програмного забезпечення. Повторне використання коду дозволяє економити час і ресурси, адже вже створені класи і методи можна використовувати у нових проєктах.

ООП допомагає знизити складність великих програмних проєктів, розділяючи їх на менші, керовані частини. Кожен об'єкт в програмі відповідає за певний набір функцій і даних, що дозволяє програмістам

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІЗ	Арк.
		Фант М.О.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

зосередитися на конкретних задачах без необхідності розуміти всю систему цілком. Це робить процес розробки більш структурованим та контрольованим.

Об'єктно-орієнтовані програми легше тестувати та налагоджувати завдяки модульній структурі. Кожен об'єкт можна тестувати окремо, що дозволяє виявляти помилки на ранніх етапах розробки. Інкапсуляція також сприяє тому, що зміни в одному об'єкті не впливають на інші об'єкти, знижуючи ймовірність виникнення нових помилок при внесенні змін.

ООП забезпечує високу гнучкість та розширюваність програм. Завдяки наслідуванню та поліморфізму, можна легко додавати нові функції та типи об'єктів без значних змін в існуючому коді. Це робить програми більш адаптивними до змін вимог та дозволяє швидше реагувати на нові виклики.

Однією з головних переваг ООП є його здатність моделювати реальний світ у програмному забезпеченні. Об'єкти можуть відповідати реальним об'єктам, таким як автомобілі, будинки чи навіть люди. Це робить програму більш зрозумілою для розробників, оскільки структура коду відображає реальні процеси та об'єкти.

Об'єктно-орієнтовані програми легше підтримувати та модернізувати, оскільки зміни можна вносити в окремі класи або об'єкти без необхідності переробляти всю систему. Це особливо важливо для довготривалих проєктів, які потребують регулярного оновлення та вдосконалення.

Об'єктно-орієнтоване програмування є потужним інструментом для створення складних програмних систем. Його основні засади — інкапсуляція, наслідування, поліморфізм та абстракція — забезпечують гнучкість, повторне використання коду, зниження складності та полегшене тестування. Завдяки цим принципам, ООП допомагає розробникам створювати більш ефективні, зрозумілі та підтримувані програми [6].

Також, в розробці використовуватиметься патерн MVC.

Модель-Вид-Контролер (MVC) — це архітектурний шаблон, який використовується для створення структурованих і масштабованих програм.

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

MVC розділяє додаток на три основні компоненти: Модель, Вид та Контролер. Цей підхід сприяє розділенню відповідальностей, що дозволяє розробникам створювати більш підтримувані і розширювані програми.

Модель відповідає за управління даними додатка. Вона взаємодіє з базою даних, відповідає за зберігання, обробку та валідацію даних. Модель не знає нічого про користувацький інтерфейс або логіку представлення, що робить її незалежною і легкою для тестування. Це дозволяє розробникам зосередитися на бізнес-логіці та забезпечити коректність даних.

Вид відповідає за відображення даних користувачу. Він отримує дані від Моделі через Контролер і форматує їх для відображення в інтерфейсі користувача. Види можуть бути простими HTML-сторінками або складними інтерфейсами, створеними за допомогою технологій, таких як JavaScript і CSS. Вид також не знає нічого про логіку обробки даних, що дозволяє розділити завдання представлення та бізнес-логіки.

Контролер є посередником між Моделлю та Видом. Він отримує запити від користувача, обробляє їх, взаємодіє з Моделлю для отримання необхідних даних і передає ці дані Виду для відображення. Контролер відповідає за бізнес-логіку додатка і забезпечує коректну обробку запитів. Це дозволяє ізолювати логіку обробки від логіки представлення і зробити код більш структурованим.

Однією з основних переваг MVC є розділення відповідальностей між різними компонентами додатка. Це дозволяє кожному компоненту зосередитися на своїх завданнях, що підвищує загальну якість і підтримуваність коду. Модель відповідає за дані, Вид — за представлення, а Контролер — за бізнес-логіку і обробку запитів.

Завдяки розділенню відповідальностей, MVC сприяє повторному використанню коду. Моделі, Види і Контролери можна використовувати в різних частинах додатка або навіть у різних проєктах. Це знижує витрати на розробку і підвищує ефективність роботи команди. Наприклад, одну і ту ж

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

Модель можна використовувати для обробки даних у різних інтерфейсах користувача.

MVC сприяє масштабованості додатків. Завдяки чіткій структурі і розділенню відповідальностей, додаток можна легко розширювати і модифікувати. Нові функції можна додавати, не змінюючи існуючу архітектуру, що знижує ризик виникнення помилок і зберігає стабільність системи. Це робить MVC ідеальним вибором для довготривалих і масштабних проєктів.

1.4 Вибір й обґрунтування інструментальних засобів для реалізації системи

При виборі засобів розробки для створення вебсайту було проведено аналіз різних мов програмування, зокрема PHP, Python, Ruby, JavaScript і Java. Кожна з цих мов має свої унікальні особливості, переваги та недоліки, що потребують детального розгляду, аби зробити обґрунтований вибір.

Однією з мов, що активно використовуються в веброботці, є Python. Ця мова відома своєю простотою та зручністю у використанні, а також великою кількістю фреймворків, таких як Django та Flask, які полегшують розробку вебзастосунків. Однак, незважаючи на переваги Python, його використання може бути обмежене в контексті деяких проєктів через більшу складність налаштування середовища виконання та меншу продуктивність у порівнянні з іншими мовами.

Ruby також заслуговує на увагу завдяки фреймворку Ruby on Rails, який забезпечує швидку та ефективну розробку вебзастосунків. Ruby on Rails відомий своєю гнучкістю та можливістю швидкого прототипування, що є важливим у початкових стадіях проєкту. Проте, варто зазначити, що продуктивність Ruby може бути нижчою у порівнянні з іншими мовами, що може стати критичним фактором при масштабуванні системи.

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

JavaScript, особливо завдяки Node.js, є потужним інструментом для серверної частини вебзастосунків. Його асинхронна природа дозволяє ефективно обробляти велику кількість запитів, що робить його чудовим вибором для реальних часів та інтерактивних вебзастосунків. Проте, JavaScript має свої складнощі, особливо у питанні управління складними структурами даних та забезпеченням безпеки.

Java є однією з найпопулярніших мов програмування завдяки своїй стабільності, надійності та масштабованості. Вона широко використовується у великих корпоративних проєктах, де критично важлива продуктивність і безпека. Однак, складність налаштування і велика кількість додаткових бібліотек та фреймворків можуть зробити розробку на Java більш трудомісткою, особливо для невеликих проєктів.

PHP було обрано як основну мову програмування для даного проєкту з кількох причин. По-перше, PHP відома своєю простотою та легкістю у вивченні, що дозволяє швидко залучати нових розробників до проєкту. Крім того, PHP має велику кількість готових рішень і бібліотек, таких як Laravel, які значно спрощують процес розробки та забезпечують високий рівень безпеки та продуктивності.

Додатково, PHP добре інтегрується з різноманітними базами даних, такими як MySQL, що забезпечує гнучкість у виборі та налаштуванні зберігання даних. PHP також підтримується на більшості вебсерверів, що робить його універсальним вибором для веброзробки. Незважаючи на критику за менш витончену архітектуру у порівнянні з іншими мовами, сучасні версії PHP демонструють високу продуктивність і стабільність, що робить його чудовим вибором для створення контентних систем управління.

Таким чином, проведений аналіз різних мов програмування дозволив визначити, що PHP є оптимальним вибором для даного проєкту завдяки своїй простоті, доступності, великій кількості готових рішень та високій продуктивності. Це дозволить забезпечити швидку та ефективну розробку

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

системи управління для вебсайту, задовольняючи всі вимоги проєкту та забезпечуючи високий рівень безпеки та надійності [7, 8].

У процесі вибору засобів розробки для бази даних для проєкту було розглянуто кілька популярних систем управління базами даних (СУБД). Серед них основними кандидатами були MySQL, PostgreSQL, SQLite, та Microsoft SQL Server. Кожна з цих СУБД має свої унікальні особливості, переваги та недоліки, які варто враховувати при виборі інструменту для конкретного проєкту.

PostgreSQL відома своєю розширюваністю та відповідністю стандартам SQL. Вона підтримує складні запити, транзакції, розширені можливості індексації та має високу продуктивність при роботі з великими обсягами даних. Однак, її налаштування і адміністрування можуть вимагати більше часу та знань, що може бути не завжди оптимальним для невеликих або середніх проєктів.

SQLite – це легка, вбудована СУБД, яка не потребує серверної частини. Вона чудово підходить для мобільних додатків, невеликих вебсайтів або прототипів, де не потрібна висока масштабованість або багатокористувацький доступ. Однак, її обмеження в управлінні одночасними з'єднаннями та транзакціями роблять її менш підходящою для масштабних або критично важливих проєктів.

Microsoft SQL Server пропонує потужні засоби для інтеграції з іншими продуктами Microsoft, такими як Azure та Power BI. Він відзначається високою продуктивністю, надійністю та розширеними функціями безпеки. Проте, його використання може бути обмежене вартістю ліцензування та вимогами до апаратних ресурсів, що може стати перешкодою для менших проєктів з обмеженим бюджетом.

Після детального аналізу було прийнято рішення використовувати MySQL як основну СУБД для проєкту. Основними причинами цього вибору стали її висока продуктивність, надійність та простота використання. MySQL підтримує широкий спектр функцій, включаючи складні транзакції,

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		21

індексацію, реплікацію та кластеризацію, що робить її універсальним рішенням для різних типів проєктів. Крім того, MySQL є відкритим програмним забезпеченням, що дозволяє зменшити витрати на ліцензування та надає можливість гнучкого налаштування відповідно до потреб проєкту [9].

Важливим аспектом при виборі MySQL стала також її висока сумісність з різними мовами програмування та фреймворками. Це забезпечує легку інтеграцію з існуючими інструментами розробки та дозволяє ефективно використовувати ресурси команди розробників. Крім того, велика спільнота користувачів MySQL надає доступ до широкого спектру ресурсів, таких як документація, форуми та бібліотеки, що полегшує вирішення можливих проблем та прискорює процес розробки.

У підсумку, вибір MySQL для проєкту був обґрунтований її технічними характеристиками, простотою адміністрування, економічною ефективністю та широкими можливостями інтеграції. Це дозволило забезпечити надійну і масштабовану базу даних, яка відповідає вимогам проєкту та сприяє його успішному впровадженню [10].

Висновки до першого розділу

Було оглянуто і проаналізовано декілька існуючих сайтів за тематикою роботи. Враховуючи їхні переваги та недоліки, було визначено вимоги до розроблюваної системи. Була обрана архітектура системи – MVC. Також, були вибрані засоби реалізації системи, а саме: мови програмування PHP, JavaScript, СУБД MySQL. Для розробки інтерфейсу застосовуватиметься бібліотека Bootstrap.

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Визначення варіантів використання системи

На рисунку (рис. 2.1) представлено алгоритм роботи розроблюваної системи у вигляді схеми.

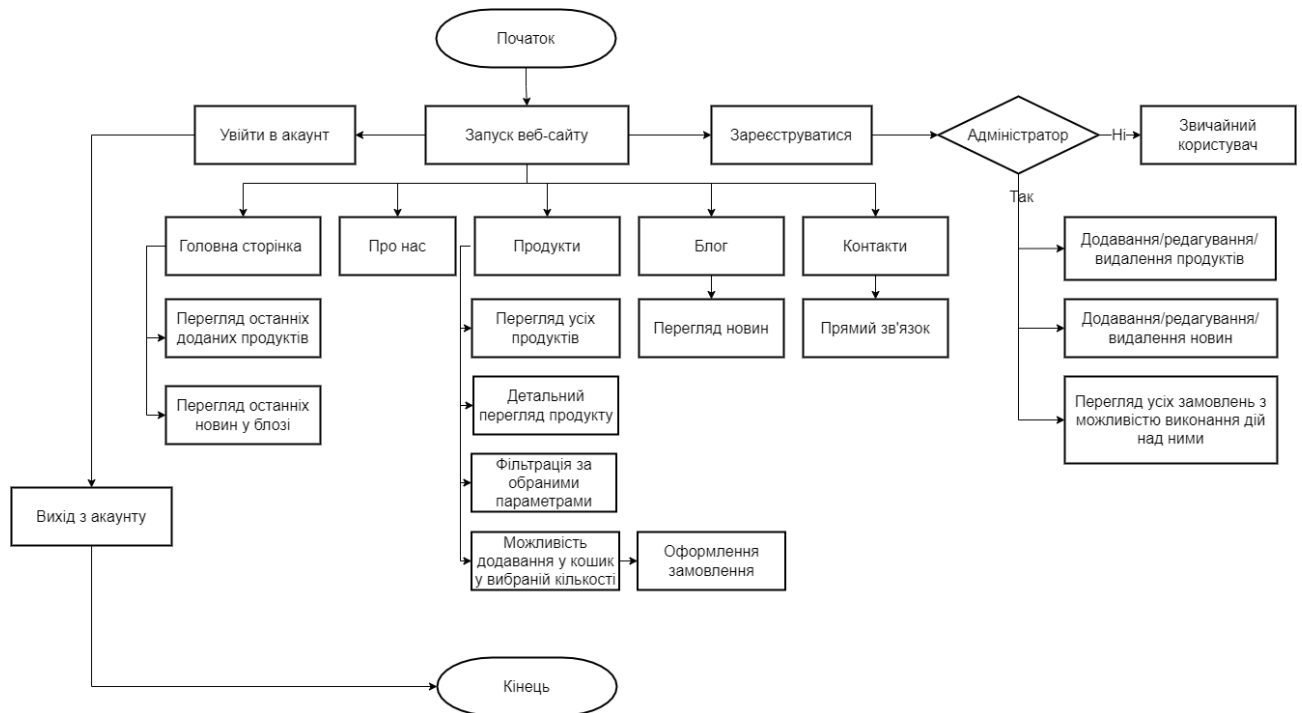


Рисунок 2.1 - Схема алгоритму роботи вебсайту

Алгоритм, представлений на рисунку (рис. 2.1), детально відтворює процес взаємодії користувачів з вебсайтом, який охоплює всі аспекти їхньої діяльності, починаючи від реєстрації до завершення сесії, а також розглядає дії адміністратора. Основний початок користувацького шляху на сайті полягає в можливості зареєструватися або увійти у вже існуючий акаунт. Це критичний момент, оскільки від нього залежать подальші доступні користувачеві опції.

Після входу на сайт користувачі потрапляють на домашню сторінку, де представлено два основних блоки інформації: огляд останніх доданих продуктів і останні новини з блогу. Ця сторінка служить вступом до контенту сайту, допомагаючи користувачам орієнтуватися в новинках і актуальних статтях. Сторінка "Про нас" містить інформацію про компанію, її місію, цілі

та ключові переваги перед конкурентами, яка дозволяє користувачам краще зрозуміти бренд і його цінності.

Однією з найважливіших частин сайту є сторінка продуктів, де користувачі можуть переглядати всі доступні товари. Тут вони можуть застосувати фільтри для відбору товарів за певними параметрами, що значно спрощує пошук потрібного продукту. Кожен товар має детальну сторінку з інформацією про характеристики, ціни, доступність та інші важливі деталі. Користувачі можуть додавати обрані товари до кошика і, в подальшому, оформити замовлення.

На сторінці блогу розміщені статті, що висвітлюють різні аспекти діяльності компанії, нові тренди у відповідній галузі або поради, що можуть бути корисними для цільової аудиторії. Контактна сторінка забезпечує засоби для зв'язку з адміністрацією через заповнення форми зворотного зв'язку, що важливо для підтримки споживчих відносин і надання допомоги в разі виникнення проблем або запитань.

Для адміністратора сайту передбачені додаткові функції, такі як управління контентом: додавання, редагування та видалення товарів та новин. Це дає змогу швидко реагувати на зміни у запасах або оновлювати інформацію відповідно до поточних потреб ринку та інтересів користувачів. Адміністратор також може переглядати всі замовлення, що були здійснені на сайті, і управляти їх статусами, підтверджуючи або скасовуючи їх відповідно до обставин.

Після завершення всіх необхідних операцій на сайті, як користувач, так і адміністратор мають можливість вийти з акаунту, що гарантує безпеку їхніх даних і завершує сесію роботи на сайті. Цей алгоритм не тільки допомагає зрозуміти структуру і логіку роботи сайту, але і планувати його подальше розширення та вдосконалення, що є ключовим для успішної діяльності в цифровий епоху.

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

2.2 Розробка бази даних

Для всіх вищевказаних можливостей та функцій створена база даних, яка зберігає потрібну інформацію (рис. 2.2).

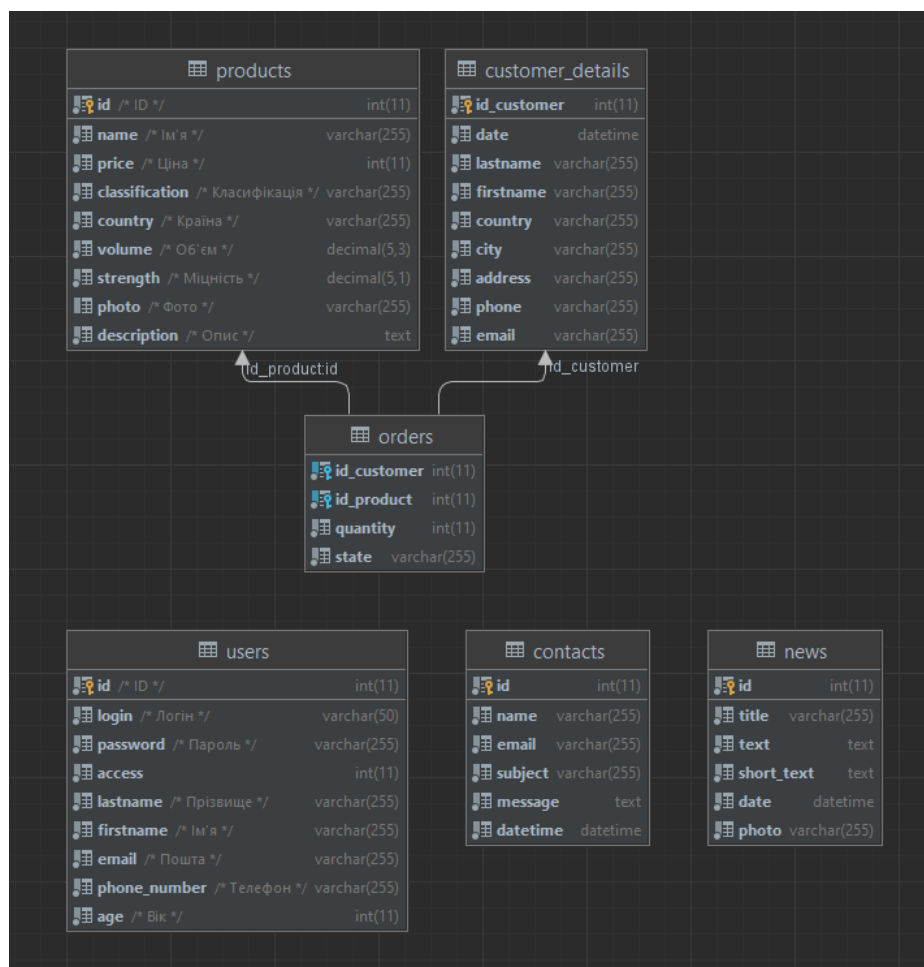


Рисунок 2.2 - Діаграма бази даних

База даних для вебсайту складається з декількох таблиць, кожна з яких має свою специфіку та функціонал. Основними таблицями є products, customer_details, orders, users, contacts та news.

Таблиця products зберігає інформацію про товари. Вона включає такі поля, як id, name (назва продукту), price (ціна), classification (класифікація), country (країна виробництва), volume (об'єм), strength (міцність), photo (фото продукту) та description (опис продукту). Ці поля дозволяють детально описати кожен продукт та зберігати всю необхідну інформацію для клієнтів.

Таблиця customer_details зберігає інформацію про клієнтів. Вона містить такі поля, як id_customer, date (дата реєстрації), lastname (прізвище),

firstname (ім'я), country (країна), city (місто), address (адреса), phone (телефон) та email (електронна пошта). Це дозволяє зберігати контактну інформацію клієнтів, що необхідна для обробки замовлень та зв'язку з ними.

Таблиця orders зберігає інформацію про замовлення. Вона включає id_customer (ідентифікатор клієнта), id_product (ідентифікатор продукту), quantity (кількість) та state (стан замовлення). Це дозволяє зберігати інформацію про те, які продукти були замовлені, у якій кількості та їх поточний стан.

Таблиця users зберігає інформацію про користувачів системи. Вона містить такі поля, як id, login (логін), password (пароль), access (рівень доступу), lastname, firstname, email, phone_number (номер телефону) та age (вік). Це дозволяє зберігати інформацію про користувачів системи та контролювати доступ до різних частин сайту.

Таблиця contacts зберігає інформацію про контактні повідомлення від користувачів. Вона включає id, name (ім'я), email, subject (тема), message (повідомлення) та datetime (дата та час повідомлення). Це дозволяє зберігати інформацію про звернення від користувачів та відповідати на них.

Таблиця news зберігає інформацію про новини. Вона містить такі поля, як id, title (заголовок), text (текст), short_text (короткий текст), date (дата публікації) та photo (фото). Це дозволяє публікувати новини та оновлення на сайті.

При створенні цієї бази даних використовуються команди SQL [11, 12, 13]. Наприклад, для створення таблиці products можна використати наступний код:

```
CREATE TABLE products (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    price INT,
    classification VARCHAR(255),
    country VARCHAR(255),
```

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

```

volume DECIMAL(5,3),
strength DECIMAL(5,1),
photo VARCHAR(255),
description TEXT
);

```

Аналогічним чином створюються й інші таблиці, кожна з яких має свій набір полів, що відповідають за зберігання відповідної інформації. Наприклад, для створення таблиці `customer_details` можна використати наступний код:

```

CREATE TABLE customer_details (
    id_customer INT AUTO_INCREMENT PRIMARY KEY,
    date DATETIME,
    lastname VARCHAR(255),
    firstname VARCHAR(255),
    country VARCHAR(255),
    city VARCHAR(255),
    address VARCHAR(255),
    phone VARCHAR(255),
    email VARCHAR(255)
);

```

Ця структура бази даних забезпечує ефективне зберігання та обробку інформації, необхідної для роботи вебсайту.

2.3 Розробка функціональних алгоритмів роботи програми

Сформувавши алгоритм роботи, можна переходити до створення програмної частини. У проєкті обов'язково має бути CMS-система та використаний архітектурний шаблон MVC. Користуючись цим патерном, проєкт складається з наступних складових: контролери, моделі та представлення. Контролери – приймають інформацію від клієнта, моделі –

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

звертаються до бази даних (при потребі), а представлення – формують HTML-сторінки, на яких можна відобразити, за допомогою переданих даних, потрібну інформацію користувачу.

Даний шаблон допомагає структурувати весь написаний код та робить його більш зрозумілим і читабельним. Також він надає можливість з легкістю розширити функціонал в майбутньому.

Далі наведена діаграма класів, на якій зображені основні класи та їх методи (рис. 2.3).

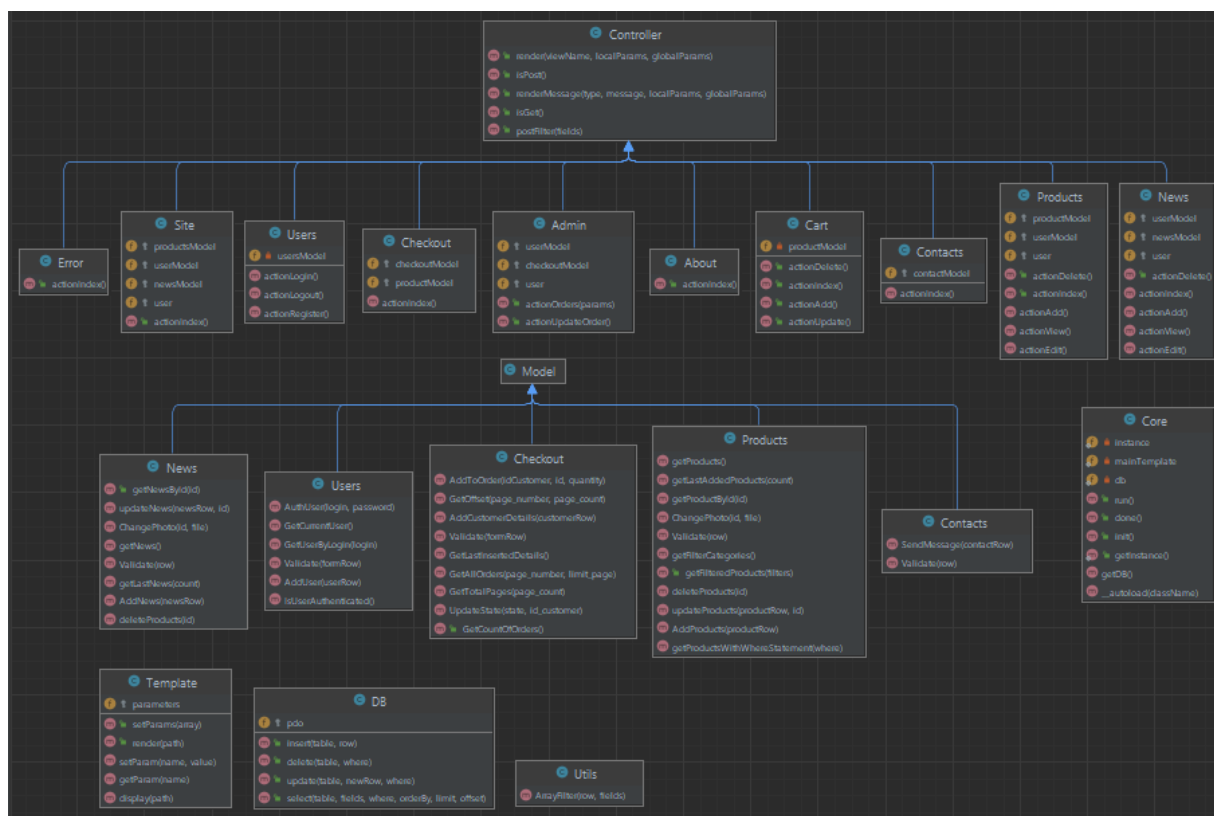


Рисунок 2.3 - Діаграма класів системи

У головному класі Core реалізовані методи, пов'язані з використанням шаблону Одинака та методи по типу getDB, що дозволяє працювати з базою даних та з методами з класу DB; run, що реалізує основний процес роботи CMS-системи та допомагає звернутися до методів відповідних контролерів, зчитуючи дані з рядка адреси; __autoload – автозавантажувача класів.

Клас Template містить методи пов'язані з встановлення параметрів (setParam, setParams), отриманням їх (getParam), підстановкою даних у представлення (render) і відображення у браузер (display).

У класі DB реалізовані методи, що використовуються в моделях, для витягування потрібної інформації з бази даних. Це такі базові методи як: select, insert, delete та update.

Клас Utils містить у собі тільки один метод – ArrayFilter, який допомагає відфільтрувати відповідний рядок витягнутий з бази даних.

У класі Controller є методи для перевірки запитів isGet та isPost, важливий метод render, в якому виконується підстановка локальних та глобальних параметрів та відображення відповідного представлення, це все виконується за допомогою раніше описаних методів з класу Template. Також є метод, що дозволяє відобразити відповідне повідомлення для користувачів, під час виконання тих чи інших дій.

Від нього наслідуються досить велика кількість класів. Контролер Products надає можливість відображення усіх продуктів – actionIndex, перегляду кожного з продуктів – actionView, додавання нових продуктів – actionAdd, редагування та видалення вже існуючих – actionEdit та actionDelete відповідно. Контролер News має доволі схожі методи, які надають той самий спектр можливостей, що й в контролері Products. В контролері Users реалізовані методи, що дозволяють зареєструватися новому користувачу – actionRegister, увійти, якщо акаунт вже існує – actionLogin та вийти з акаунту – actionLogout. В контролері Admin реалізовані два методи, перший з яких actionOrders, що дозволяє побачити всі замовлення та actionUpdateUsers, який викликається асинхронно та змінює стан замовлень. Контролер Cart, має методи, що пов'язані з видаленням, додаванням та переглядом кошику й метод оновлення кількості продуктів у кошику, що також викликається асинхронно. Контролер Error допомагає відобразити помилку при неправильно введеній URL-адресі. Такі контролери як About, Contacts та Site просто візуалізують відповідні сторінки, а в контролері Contacts, також приймають дані з форми і викликають відповідний метод моделі, для відправки повідомлення і створення запису у базі даних.

У базовому класі Model не реалізовано ніяких методів.

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

Проте від нього наслідуються 5 моделей. У моделі Users передбачені методи додавання в базу даних нового користувача при реєстрації, авторизації, отримання поточного користувача та користувача за логіном, перевірка на автентифікацію і метод валідації. У моделі Products є методи, що дозволяють змінити/встановити фото для продукту, додавання продуктів, їх видалення та редагування. Також ряд методів, які використовуються для отримання потрібних даних з таблиці products. У моделі News приблизно методи з тією ж самою логікою, тому детально описувати їх немає сенсу. Модель Contacts містить два методи: метод відправки повідомлення на пошту компанії і додавання його до бази даних та метод валідації. Остання з 5-ти моделей, модель Checkout відповідає за додавання деталей замовлення і самих замовлень до бази даних, їх оновлення, та валідації, а також містить ряд методів, що використовуються для пагінації, на сторінці замовлень у адміністратора.

2.4 Реалізація системи

Ознайомившись зі складовими вебсайту та проаналізувавши діаграму класів, слід описати основні методи і функції, які були реалізовані в цих класах.

В першу чергу розглянемо клас Controller.

Клас Controller призначений для обробки HTTP-запитів і рендерингу представлень у вебдодатку. Він містить методи, які допомагають визначити тип запиту, фільтрувати дані, рендерити представлення та повідомлення.

```
public function isPost()
{
    if($_SERVER['REQUEST_METHOD'] == 'POST')
        return true;
    else
        return false;
}
```

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public function isGet()
{
    if($_SERVER['REQUEST_METHOD'] == 'GET')
        return true;
    else
        return false;
}

```

Метод, isPost, перевіряє, чи є HTTP-запит методом POST. Якщо \$_SERVER['REQUEST_METHOD'] дорівнює 'POST', метод повертає true, в іншому випадку - false. Аналогічно працює метод isGet, який перевіряє, чи є запит методом GET, і повертає відповідний результат.

```

public function postFilter($fields)
{
    return Utils::ArrayFilter($_POST, $fields);
}

```

Метод postFilter використовується для фільтрації даних з масиву \$_POST. Він приймає масив полів, які потрібно відфільтрувати, і використовує утиліту ArrayFilter з класу Utils для обробки даних. Це дозволяє виділити необхідні поля з POST-запиту для подальшого використання у додатку.

```

public function render($viewName, $localParams = null, $globalParams = null)
{
    $tpl = new Template();
    if(is_array($localParams))
        $tpl->setParams($localParams);
    if(!is_array($globalParams))
        $globalParams = [];
    $moduleName = strtolower((new \ReflectionClass($this))->getShortName());
    $globalParams['PageContent'] =
        $tpl->render("app/views/{$moduleName}/{ $viewName }.php");
    return $globalParams;
}

```

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				31
Змн.	Арк.	№ докум.	Підпис	Дата		

Метод `render` відповідає за рендеринг представлення на основі заданого імені представлення і параметрів. Спочатку він створює об'єкт `Template` і передає локальні параметри у шаблон, якщо вони є масивом. Далі, за допомогою рефлексії, отримується ім'я модуля, що дозволяє визначити шлях до представлення. Після цього представлення рендериться, а його вміст додається до глобальних параметрів під ключем `'PageContent'`, які потім повертаються.

```
public function renderMessage($type, $message, $localParams = null, $globalParams = null)
{
    $tpl = new Template();
    if(is_array($localParams))
        $tpl->setParams($localParams);
    $tpl->setParam('MessageText', $message);
    switch ($type)
    {
        case 'success':
            $tpl->setParam('MessageClass', 'success');
            break;
        case 'error':
            $tpl->setParam('MessageClass', 'danger');
            break;
        case 'info':
            $tpl->setParam('MessageClass', 'info');
            break;
    }
    if(!is_array($globalParams))
        $globalParams = [];
    $globalParams['PageContent'] = $tpl->render("app/views/layout/message.php");
    return $globalParams;
}
```

Метод `renderMessage` використовується для рендерингу повідомлень з заданим типом і текстом. Він приймає тип повідомлення (`success`, `error`, `info`),

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				32
Змн.	Арк.	№ докум.	Підпис	Дата		

текст повідомлення, локальні та глобальні параметри. Як і у методі render, об'єкт Template створюється і налаштовується за допомогою локальних параметрів. Далі встановлюється клас повідомлення на основі його типу, а саме повідомлення рендериться і додається до глобальних параметрів під ключем 'PageContent', які потім повертаються.

Клас Controller забезпечує гнучкість і зручність у роботі з HTTP-запитами та рендерингом представлень у вебдодатку, спрощуючи обробку запитів і створення динамічних сторінок.

Також, одним з головних класів є клас Core. Він забезпечує централізоване управління основними процесами CMS, включаючи ініціалізацію, обробку запитів, рендеринг і автозавантаження класів. Розглянемо його основні методи.

```
private function __construct()
{
    global $Config;
    spl_autoload_register('\core\Core::__autoload');
    self::$db = new DB($Config['Database']['Server'],
        $Config['Database']['Username'],
        $Config['Database']['Password'],
        $Config['Database']['Database']);
}
```

Метод __construct є приватним і викликається лише один раз через метод getInstance. Він ініціалізує глобальні налаштування, реєструє автозавантажувач класів і встановлює з'єднання з базою даних, використовуючи параметри конфігурації.

```
public static function getInstance()
{
    if(empty(self::$instance)) {
        self::$instance = new Core();
        return self::getInstance();
    }
    else
```

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				33
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return self::$instance;
    }

```

Метод `getInstance` реалізує логіку `Singleton`. Якщо екземпляр класу ще не створений, він створює новий екземпляр і повертає його, інакше повертає існуючий екземпляр.

```

public function init()
{
    session_start();
    self::$mainTemplate = new Template();
}

```

Метод `init` відповідає за ініціалізацію системи. Він запускає сесію і створює новий екземпляр класу `Template`, який буде використовуватися для рендерингу шаблонів.

```

public function run()
{
    $path = $_GET['path'];
    $pathParts = explode('/', $path);
    $className = ucfirst($pathParts[0]);
    if(empty($className))
        $fullClassName = 'controllers\\Site';
    else
        $fullClassName = 'controllers\\'.$className;
    $methodName = ucfirst($pathParts[1]);
    if(empty($methodName))
        $fullMethodName = 'actionIndex';
    else
        $fullMethodName = 'action'.$methodName;
    if(class_exists($fullClassName)) {
        $controller = new $fullClassName();
        if (method_exists($controller, $fullMethodName)){
            $method = new \ReflectionMethod($fullClassName, $fullMethodName);
            $paramsArray = [];
            foreach($method->getParameters() as $parameter)
            {

```

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        array_push($paramsArray,      isset($_GET[$parameter->name])      ?
$_GET[$parameter->name] : null);
    }
    $result = $method->invokeArgs($controller, $paramsArray);
    if(is_array($result))
    {
        self::$mainTemplate->setParams($result);
    }
}
else
    header('Location: /error/');
    // throw new \Exception('404 Not Found');
} else
    header('Location: /error/');
    // throw new \Exception('404 Not Found');
}

```

Метод run виконує основний процес роботи CMS. Він обробляє URL-запит, розбиваючи його на частини, щоб визначити ім'я класу контролера і метод, який необхідно викликати. Якщо клас контролера існує і метод доступний, він викликає метод з відповідними параметрами. Якщо клас або метод не існують, користувача перенаправляють на сторінку помилки.

```

public static function __autoload($className)
{
    $fileName = 'app\\'.$className.'.php';
    if(is_file($fileName))
        include($fileName);
}

```

Метод __autoload є статичним і відповідає за автозавантаження класів. Він приймає ім'я класу, додає до нього шлях і розширення .php, а потім включає цей файл, якщо він існує.

Клас Core забезпечує централізоване управління основними процесами CMS, включаючи ініціалізацію, обробку запитів, рендеринг і автозавантаження класів.

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				35
Змн.	Арк.	№ докум.	Підпис	Дата		

Розглянемо клас, що відповідає за роботу з користувачами, а саме модель User. Клас успадковується від базового класу Model та надає функціонал для управління користувачами в системі. Він включає методи для валідації даних користувачів, аутентифікації, додавання нових користувачів, перевірки автентичності, отримання поточного користувача і пошуку користувачів за логіном.

```
public function Validate($formRow)
{
    $errors = [];
    if(empty($formRow['login']))
        $errors [] = 'Поле "Логін" не може бути порожнім';
    $user = $this->GetUserByLogin($formRow['login']);
    if(!empty($user))
        $errors [] = 'Користувач з вказаним логіном вже зареєстрований';
    if(empty($formRow['password']))
        $errors [] = 'Поле "Пароль" не може бути порожнім';
    if(strlen($formRow['password']) < 6)
        $errors [] = 'Поле "Пароль" не може бути коротше ніж 6 символів';
    if($formRow['password'] != $formRow['repeatedPassword'])
        $errors [] = 'Паролі не збігаються';
    if(empty($formRow['firstname']))
        $errors [] = 'Поле "Ім'я" не може бути порожнім';
    if(empty($formRow['lastname']))
        $errors [] = 'Поле "Прізвище" не може бути порожнім';
    if(empty($formRow['email']))
        $errors [] = 'Поле "Пошта" не може бути порожнім';
    if(empty($formRow['phone_number']))
        $errors [] = 'Поле "Номер телефону" не може бути порожнім';
    if(empty($formRow['age']))
        $errors [] = 'Поле "Вік" не може бути порожнім';
    if($formRow['age'] < 18)
        $errors [] = 'Вам має бути не менше 18 років, для реєстрації';
    if(count($errors) > 0)
        return $errors;
```

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				36
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        else
            return true;
    }

```

Метод Validate перевіряє вхідні дані з форми реєстрації на наявність помилок. Він перевіряє, чи заповнені обов'язкові поля, чи унікальний логін, чи збігаються паролі, чи дотримуються мінімальні вимоги до довжини пароля і чи відповідає вік користувача вимогам (не менше 18 років). Якщо знаходить помилки, повертає масив з повідомленнями про помилки, інакше повертає true.

```

public function IsUserAuthenticated()
{
    return isset($_SESSION['user']);
}

```

```

public function GetCurrentUser()
{
    if($this->IsUserAuthenticated())
        return $_SESSION['user'];
    else
        return null;
}

```

Метод IsUserAuthenticated перевіряє, чи є користувач автентифікованим. Він повертає true, якщо сесія користувача активна, інакше false.

Метод GetCurrentUser повертає дані поточного автентифікованого користувача. Якщо користувач автентифікований, повертає дані користувача з сесії, інакше повертає null.

```

public function AddUser($userRow)
{
    $validateResult = $this->Validate($userRow);
    if(is_array($validateResult))
        return $validateResult;
    $fields = ['login', 'password', 'firstname', 'lastname', 'email', 'phone_number', 'age'];
}

```

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				37
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    $userRowFiltered = Utils::ArrayFilter($userRow, $fields);
    $userRowFiltered['password'] = md5($userRowFiltered['password']);
    \core\Core::getInstance()->getDB()->insert('users', $userRowFiltered);
    return true;
}

```

Метод AddUser додає нового користувача до бази даних. Він спочатку перевіряє дані користувача за допомогою методу Validate. Якщо валідація проходить успішно, фільтрує вхідні дані, хешує пароль за допомогою md5 і зберігає користувача в базу даних. Якщо валідація не проходить, повертає масив з помилками.

```

public function AuthUser($login, $password)
{
    $password = md5($password);
    $users = \core\Core::getInstance()->getDB()->select('users', '*', [
        'login' => $login,
        'password' => $password
    ]);
    if(count($users) > 0)
    {
        $user = $users[0];
        return $user;
    } else
    {
        return false;
    }
}

```

Метод AuthUser виконує аутентифікацію користувача за логіном і паролем. Він хешує пароль, шукає користувача в базі даних з відповідними логіном і паролем. Якщо користувач знайдений, повертає його дані, інакше повертає false.

```

public function GetUserByLogin($login)
{
    $rows = \core\Core::getInstance()->getDB()->select('users', '*', ['login' => $login]);
    if (count($rows) > 0)
    {
        return $rows[0];
    }
}

```

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

```

else
    return null;
}

```

Метод `GetUserByLogin` повертає дані користувача за логіном. Він шукає користувача в базі даних за логіном. Якщо користувач знайдений, повертає його дані, інакше повертає `null`.

Клас `Users` забезпечує основні операції з користувачами, такі як реєстрація, аутентифікація, валідація даних і отримання інформації про користувачів. Цей функціонал є ключовим для будь-якої системи, яка працює з користувачами і потребує управління їх обліковими записами та доступом.

Висновки до другого розділу

Було представлено та детально описано алгоритм роботи вебсайту. Також, було спроектовано та розроблено базу даних, яка складається з шести таблиць. Окрім цього, була розглянута діаграма класів із детальним описом ролі кожного з них.

До того ж, був проведений детальний аналіз основних класів і описано методи, які в них реалізовані, з вказанням ролі, яку вони відіграють у загальному алгоритмі веборієнтованої системи.

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				39
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБОРІЄНТОВАНОЮ СИСТЕМОЮ

3.1 Структура інтерфейсу системи

Зайшовши на сайт, користувача зустрічає головна сторінка, на якій розміщений слайдер, новинки та останні новини з блогу. Доступний хедер сайту, на якому розміщені сторінки сайту: «Головна», «Про нас», «Продукти», «Контакти». Також є можливість переглянути корзину та зареєструватись(рис. 3.1).

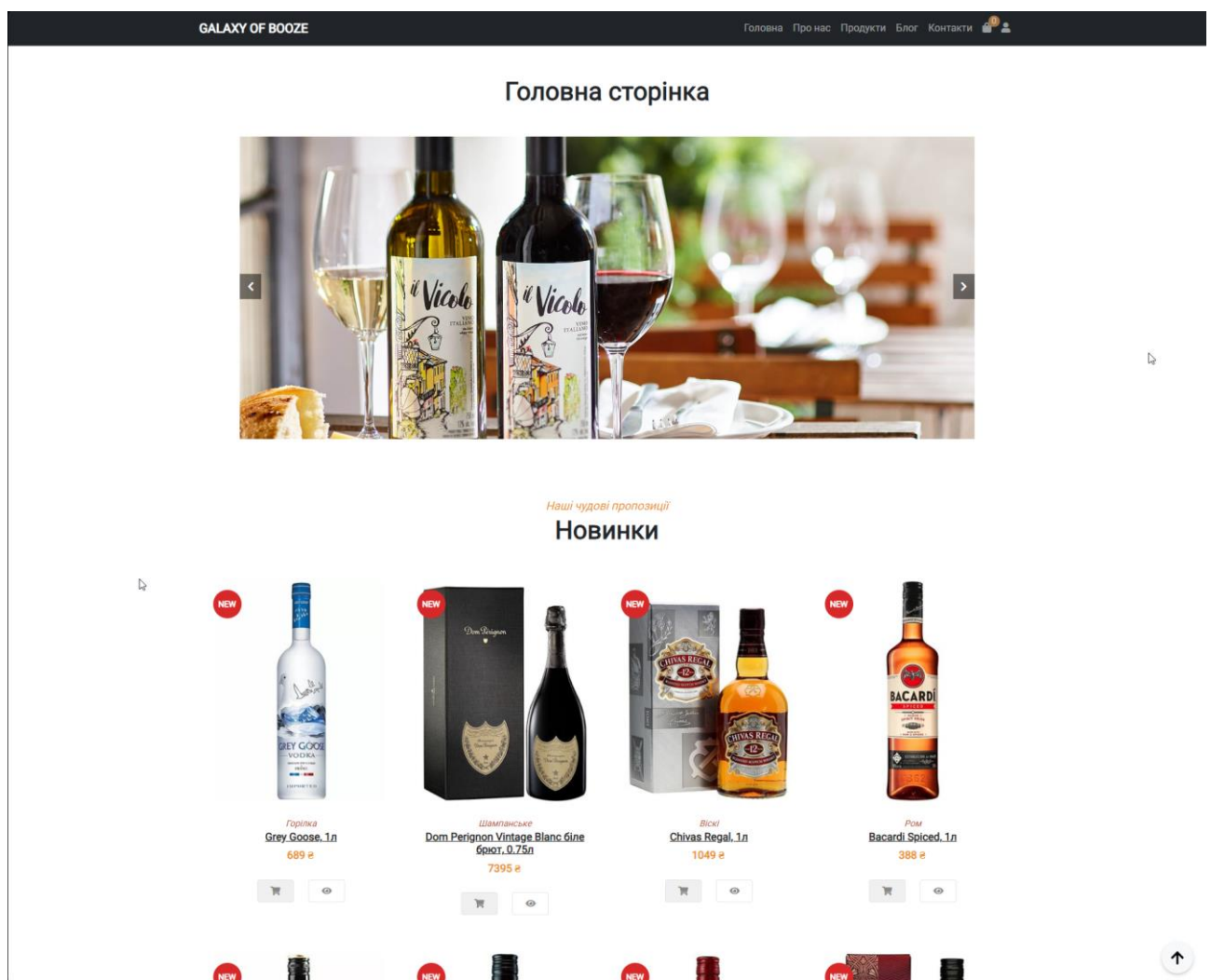


Рисунок 3.1 - Головна сторінка

Наступною є сторінка Про нас, яка ознайомлює користувача з інформацією про компанію, її цілями та пріоритетами(рис. 3.2).

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

Про нас



ОНЛАЙН-ПІДТРИМКА 24/7
Ви можете звернутись до служби підтримки в будь-який момент часу



ГАРАНТІЯ ПОВЕРНЕННЯ ГРОШЕЙ
Ви можете бути впевнені в тому, що ваші гроші будуть повернуті, у випадку повернення товару



БЕЗКОШТОВНА ДОСТАВКА ТА ПОВЕРНЕННЯ
Доставка та повернення - за наш рахунок



З 2003 року

Про наш інтернет-магазин

Galaxy of booze є світовим лідером у продажі алкогольних напоїв із видатною колекцією брендів алкогольних напоїв – бізнес, побудований на принципах та засадах, закладених гігантами галузі.

Наші амбіції щодо продуктивності – бути однією з найбільш ефективних, найнадійніших і шанованих компаній у світі.

18 років досвіду у бізнесі

Рисунок 3.2 - Сторінка «Про нас»

Далі – основна сторінка, де розміщуються всі продукти та фільтри. Можна фільтрувати за ціною товару та/або його типом: шампанське, віскі, ром, шампанське, лікер, горілка (рис. 3.3).

Оберіть фільтри для продуктів:

Мін. ₴ - Макс. ₴

Nothing selected

Горілка
Grey Goose, 1л
689 ₴

Шампанське
Dom Perignon Vintage Blanc
біле брют. 0.75л
7395 ₴

Віскі
Chivas Regal, 1л
1049 ₴

Ром
Bacardi Spiced, 1л
388 ₴

Рисунок 3.3 - Сторінка продуктів

Кожен продукт можна окремо переглянути, ознайомитися з його характеристиками: ціна, тип, опис, рисунок, та замовити у вибраній кількості (рис. 3.4).

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

Jim Beam Apple 0.7 л 32.5%



Країна: [США](#) Класифікація: [Лікер](#)

479 ₴

Історія компанії **Jim Beam**, широко відомої у всьому світі, пов'язана з ім'ям Джейкоба Біма, який наприкінці 18 століття почав виробляти віскі з розмелених кукурудзяних зерен, жита та солоду. Цей міцний спиртний напій бурштинового кольору отримав назву бурбон за назвою округу Бурбон у штаті Кентуккі, де він був уперше зроблений. Джейкоб Бім продав першу діжку свого бурбона у 1795 році. У наш час історія компанії Jim Beam налічує вже понад 200 років, але власники компанії все також продовжують слідувати багатим традиціям своїх предків, передаючи рецепт чудового віскі з покоління до покоління.

"Jim Beam" - бурбон №1 у світі! Щороку у всьому світі випивається понад 40 мільйонів літрів цього дивовижного напою. Це справжній віскі, який отримав повсюдне визнання завдяки своїй відмінній якості та веселому характеру. Не менше 51% сула, за прийнятими стандартами, створюється із кукурудзи. "Джим Бім" витримують протягом 4-х років у нових обвуглених бочках, виготовлених з деревини американського дуба.

Add to Cart



Рисунок 3.4 - Сторінка товару

Після того, як було додано товар у кошик, можна побачити обрані товари, ціну всього замовлення та змінити кількість бажаних товарів, або прибрати замовлення з кошика(рис. 3.5).

Продукти	Ціна	Кількість	Повна ціна	
 Jim Beam Apple Лікер	479 ₴	3	1437 ₴	x
 Bacardi Spiced Ром	388 ₴	2	776 ₴	x

Усього в кошику

Всього: 2213 ₴

Оформити замовлення



Рисунок 3.5 - Кошик товарів

Після цього користувач може заповнити форму, ввівши відповідні дані для замовлення. Дана реалізація передбачає, що замовлення може бути подарунковим, тому потрібно увести бажані дані(рис. 3.6).

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

Оформлення замовлення

Ім'я

Ім'я

Прізвище

Прізвище

Країна

Країна

Адреса

Адреса

Місто

Місто

Телефон

Телефон

Пошта

Пошта

Усього в кошику

Всього: 2601 ₴

Оформити

Рисунок 3.6 - Оформлення замовлення

Також на сайті є можливість переглянути блог, у якому містяться новини. Новини відповідають тематиці сайту. Присутня дата публікації новини, рисунок до неї, можливість повністю переглянути новину (рис. 3.7).

Останні новини



2024-06-06 13:34:50

Корисні властивості вина обумовлені не алкоголем

Так вважає команда британських учених...

Читати далі



2024-06-05 13:00:12

Сатурналії - давньоримський аналог новорічних корпоративів

Святкувались вони із 17 грудня...

Читати далі



2024-06-05 12:28:13

П'ять фактів про текилу Дуейна Джонсона

Текила, зростанню популярності якої не завадив ковід...

Читати далі



2024-06-05 12:15:33

У Великій Британії вдесьяте провели "тверезий січень"

У 2014 році у Великій Британії стартувала акція "Тверезий січень"...

Читати далі

Рисунок 3.7 - Сторінка блогу

Останнім пунктом меню є сторінка «Контакти», на якій розміщена контактна інформація та форма, з допомогою якої можна зв'язатися з адміністрацією (рис. 3.8).

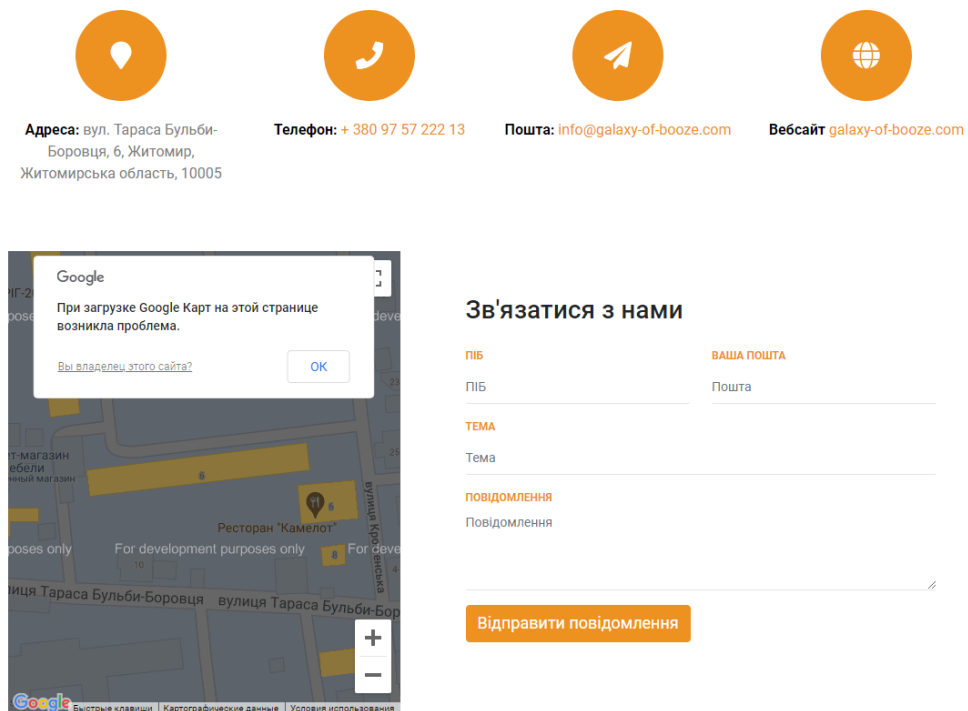


Рисунок 3.8 - Сторінка контактів

Також, зайшовши на сайт, як адміністратор, у користувача є можливість додати/видалити/відредагувати продукти та новини (рис. 3.9).

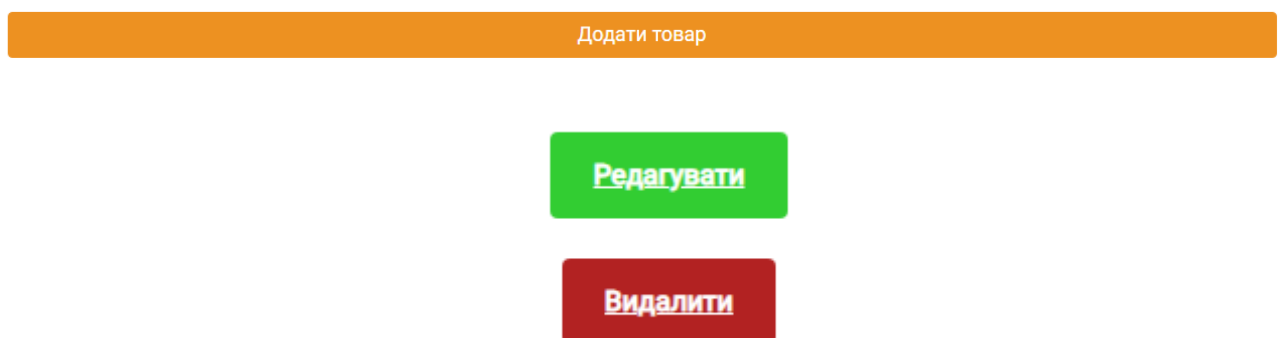


Рисунок 3.9 - Кнопки керування контентом

Всі замовлення адміністратор може переглянути на окремій сторінці, де дані виведені в таблицю і реалізована пагінація. Переглядаючи таблицю з замовленнями, адміністратор або погоджує замовлення, або ні. Відповідний колір буде представлено у таблиці(рис. 3.10).

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

Замовлення

Прізвище та ім'я замовника	Замовлені товари та їх кількість	Телефон	Дата замовлення	Дії
Тест Тест	Chivas Regal: 1	0999999999	2024-06-13 18:20	Оформлено
Тест1 Тест1	Baileys Original: 1	0999999999	2024-06-13 18:20	Оформлено
Тест2 Тест2	Bacardi Spiced: 1	0999999999	2024-06-13 18:20	Оформлено
Тест3 Тест3	Hankey Bannister Original: 1	0999999999	2024-06-13 18:20	Оформлено
Тест4 Тест4	Jameson Irish Whiskey: 1	0999999999	2024-06-13 18:20	Оформлено
Тест5 Тест5	Jim Beam Apple: 1	0999999999	2024-06-13 18:20	Відхилено
Тест6 Тест6	Chivas Regal: 1	0999999999	2024-06-13 18:20	Оформлено
Тест7 Тест7	Jameson Irish Whiskey: 1	0999999999	2024-06-13 18:20	Оформлено
Тест9 Тест9	Baileys Original: 1	0999999999	2024-06-13 18:20	Відхилено
Тест10 Тест10	Dom Perignon Vintage Blanc біле брют / Bacardi Spiced: 1 / 1	0999999999	2024-06-13 18:20	Оформлено

Рисунок 3.10 - Перегляд усіх замовлень

3.2 Тестування системи

Важливим етапом в розробці будь-якого великого проєкту є його тестування.

Основна помилка, яка може виникати – це спроба звичайного користувача, виконати дії, які може виконувати тільки адміністратор. Ця помилка оброблюється у роботі і користувач отримує відповідне повідомлення (рис. 3.11)

Доступ заборонено

Доступ до даної сторінки можуть мати тільки користувачі з відповідними правами доступу

Рисунок 3.11 - Перегляд усіх замовлень

Також, якщо користувач помилково, або навмисно введе в рядок адреси неіснуючу адресу, то він буде перенаправлений на сторінку 404 (рис. 3.11).

Помилка 404

Сторінку не знайдено

Неправильно набрано адресу або такої сторінки на сайті більше не існує.

[Перейти на головну сторінку](#)

Рисунок 3.12 - Помилка 404

При намаганнях оформити замовлення без товарів в кошику, користувачу буде виведено відповідне повідомлення та посилання на сторінку товарів, для їх вибору (рис. 3.13).

Оформлення замовлення

Ваш кошик пустий

[Товари](#)

Рисунок 3.13 - Обробка помилки, коли оформлення відбувається без товарів в кошику

Далі перевіримо стандартну процедуру оформлення замовлення. Оберемо товар, додамо в кошик, замовимо товар.

Кошик

Продукти					Ціна	Кількість	Повна ціна	
		Chivas Regal Bickl			1049 ₪	1	1049 ₪	×

Рисунок 3.14 - Товар в кошику

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ		Арк.
		Фант М.О.					46
Змн.	Арк.	№ докум.	Підпис	Дата			

GALAXY OF BOOZE

ГоловнаПронасПродуктиБлогКонтакти

Оформлення замовлення

Ім'я

Тест

Прізвище

Тест

Країна

Україна

Адреса

вул Київська, 6

Місто

Житомир

Телефон

0999999999

Пошта

test@test.test

Усього в кошику

Всього:1049 ₴

Оформити

З'яжіться з нами в соціальних мережах:

f

t

G

Рисунок 3.15 - Оформлення замовлення

Перевіримо наявність замовлення в CRM.

GALAXY OF BOOZE

ГоловнаПронасПродуктиБлогКонтактиtestВийти

Замовлення

Прізвище та ім'я замовника	Замовлені товари та їх кількість	Телефон	Дата замовлення	Дії
Тест Тест	Chivas Regal: 1	0999999999	2024-06-13 18:20	✓✗

Рисунок 3.16 - Замовлення в CRM

Можемо побачити демонстрацію кнопок для адміністратора.

GALAXY OF BOOZE

ГоловнаПронасПродуктиБлогКонтактиtestВийти

Замовлення

Прізвище та ім'я замовника	Замовлені товари та їх кількість	Телефон	Дата замовлення	Дії
Тест Тест	Chivas Regal: 1	0999999999	2024-06-13 18:20	Оформлено

Рисунок 3.17 - Підтвердження замовлення в CRM

По проведеним тестам можна визначити, що система замовлень працює відмінно, не маючи жодних функціональних помилок.

Далі перевіримо роботу реєстрації на сайті

Реєстрація на сайті

Логін

Пароль

Пароль (ще раз)

Прізвище

Ім'я

Пошта

Телефон

Вік

☐ Я приймаю [Умови використання](#) та [Політику конфіденційності](#)

Рисунок 3.18 - Форма реєстрації

Вік

☐ Я приймаю [Умови використання](#) та [Політику](#)

 Поставте тут прапорець, якщо хочете продовжити.

Вже маєте акаунт? [Увійти](#)

Рисунок 3.19 - Помилка, якщо не погодитись з умовами та політками сайта

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				48
Змн.	Арк.	№ докум.	Підпис	Дата		

Пошта

test

Електронна адреса має містити знак "@". В електронній адресі "test" знака "@" немає.

1234567890

Рисунок 3.20 - Помилка, якщо ввести неправильний формат пошти

Поле "Пароль" не може бути коротше ніж 6 символів

Логін

test

Пароль

Пароль

Пароль (ще раз)

Пароль (ще раз)

Рисунок 3.21 - Помилка, якщо ввести закороткий пароль

GALAXY OF BOOZE

Головна Про нас Продукти Блог Контакти

Реєстрація на сайті

Користувач успішно зареєстрований

Рисунок 3.21 - Результати успішної реєстрації

Перевіримо роботу форми входу на сайт.

З поверненням

Логін

test

Пароль

.....

Увійти

Ще не зареєструвалися? [Зареєструватися](#)

Рисунок 3.22 - Форма входу в обліковий запис

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				49
Змн.	Арк.	№ докум.	Підпис	Дата		

Вхід на сайт

Ви успішно ввійшли в акаунт

Рисунок 3.23 - Результат успішної авторизації

Підвищимо рівень доступу нового користувача, щоб перевірити роботу редагування товарів. До підвищення немає кнопок «Редагувати» та «Видалити» (рис. 3.24)

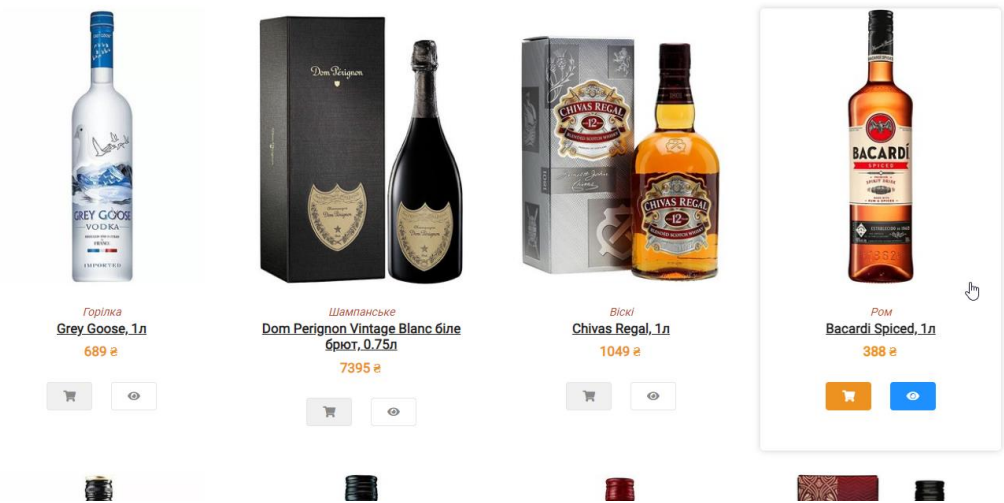


Рисунок 3.24 - Інтерфейс сайту до підвищення рівня доступу користувача

Кнопки редагування з'явилися (рис. 3.25)



Рисунок 3.25 - Інтерфейс сайту після підвищення рівня доступу користувача

З зображень видно, що користувачу стали доступними функції редагування та видалення товарів. Підвищимо ціну першого напою зі списку на 100 гривень.

Назва товару

Grey Goose

Ціна

789

Класифікація

Горілка

Країна

Франція

Об'єм

1,000

Міцність

40,0

Фото

Вибрати файл

Файл не вибрано



Рисунок 3.26 - Інтерфейс редагування товару

Можна побачити, що ціна обраного напою змінилась. Отже, все працює коректно.



Горілка
Grey Goose, 1л
789 ₴

Редагувати

Видалити



Шампанське
Dom Perignon Vintage Blanc біле
брют, 0.75л
7395 ₴

Редагувати

Видалити



Віскі
Chivas Regal, 1л
1049 ₴

Редагувати

Видалити



Ром
Bacardi Spiced, 1л
388 ₴

Редагувати

Видалити

Рисунок 3.27 - Результат редагування товару

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				51
Змн.	Арк.	№ докум.	Підпис	Дата		

GALAXY OF BOOZE

[Головна](#)
[Про нас](#)
[Продукти](#)
[Блог](#)
[Контакти](#)

test

[Вийти](#)

Додати новину		
	📅 2024-06-06 Корисні властивості вина обумовлені не алкоголем Так вважає команда британських учених... <i>Читати далі</i> Редагувати Видалити	
	📅 2024-06-05 Сатурналії - давньоримський аналог новорічних корпоративів Святкувались вони із 17 грудня... <i>Читати далі</i> Редагувати Видалити	

Заголовок новини

Тестова новина

Короткий текст новини

Параграф

B *I*         

▼

⋮

Тестова новина

Повний текст новини

Параграф

B *I*         

▼

⋮

Тестова новина

Фото

Вибрати файл

11_Alpine-A110-San-Remo-7...sie-voor-rallysport-scaled.jpg

Зберегти

		Коломієць А.О.			ІІЗ.КР.Б-121-24-ІІЗ	Арк.
		Фант М.О.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

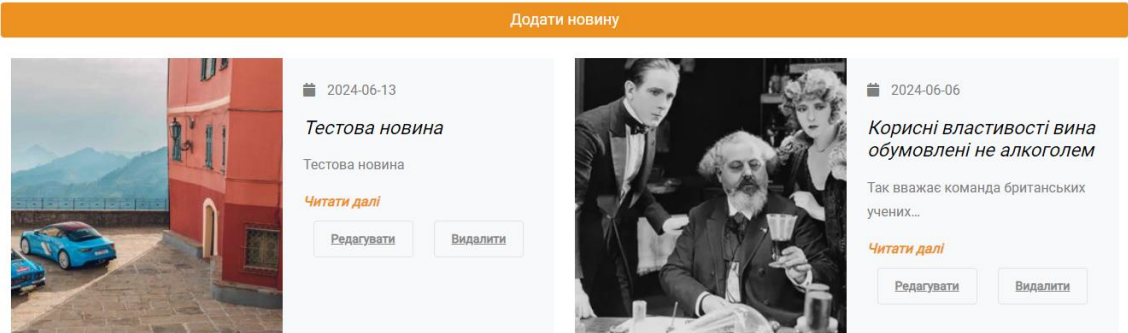


Рисунок 3.30 - Новостворена новина

Загалом, можна відзначити, що вебсайт успішно пройшов усі перевірки та тести. Жоден з функціональних елементів не мав помилок та працював відмінно.

Висновки до третього розділу

Підводячи підсумки, можна зазначити, що було детально розглянуто інтерфейс вебсайту і детально описано сторінки, які відображаються при роботі з ним.

Також сайт було протестовано і висвітлено основні помилки, які в теорії могли б виникнути при користуванні та як вони оброблюються.

ВИСНОВКИ

У випускній кваліфікаційній роботі бакалавра розроблено веборієнтовану систему для вибору та замовлення напоїв.

Було оглянуто і проаналізовано декілька існуючих сайтів за тематикою роботи. Враховуючи їхні переваги та недоліки, було визначено вимоги до розроблюваної системи. Була обрана архітектура системи – MVC. Також, були вибрані засоби реалізації системи, а саме: мови програмування PHP, JavaScript, СУБД MySQL. Для розробки інтерфейсу застосована бібліотека Bootstrap.

Було представлено та детально описано алгоритм роботи вебсайту. Також, було спроектовано та розроблено базу даних, яка складається з шести таблиць. Окрім цього, була розглянута діаграма класів із детальним описом ролі кожного з них.

До того ж, був проведений детальний аналіз основних класів і описано методи, які в них реалізовані, з вказанням ролі, яку вони відіграють у загальному алгоритмі веборієнтованої системи.

Підводячи підсумки, можна зазначити, що було розроблено інтерфейс вебсайту, який відповідає встановленим вимогам, а саме: зручний та привабливий; мова – українська; зручна система для сортування та фільтрації напоїв.

Також, сайт успішно пройшов декілька етапів тестування основних функціональних модулів системи. Було протестовано найбільш теоретично вразливі модулі, такі як реєстрації користувачів, входу в систему, розділення користувачів на ролі, доступу користувачів до різних складових системи залежно від їхнього рівня доступу та додавання або редагування нових товарів. Крім того, було перевірено систему оформлення та обробки замовлень.

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				54
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. goodwine.com.ua [Електронний ресурс] – Режим доступу до ресурсу: goodwine.com.ua.
2. vintagemarket.com.ua [Електронний ресурс] – Режим доступу до ресурсу: vintagemarket.com.ua.
3. elitochka.net [Електронний ресурс] – Режим доступу до ресурсу: elitochka.net.
4. Сесіл Мартін Р. Чиста архітектура / Роберт Сесіл Мартін., 2019. – 368 с.
5. Сесіл Мартін Р. Чистий код: створення, аналіз, рефакторинг / Роберт Сесіл Мартін., 2019. – 416 с.
6. Плюси і мінуси об'єктно-орієнтованого програмування [Електронний ресурс] - 2015. Режим доступу до ресурсу: <http://damp.biz/plyusi-i-minusi-obyektno-oriyentovanogo-programuvannya/>
7. Matt Zandstra. PHP Objects, Patterns, and Practice, [Електронний ресурс] - 2013. Режим доступу до ресурсу: <https://www.apress.com>
8. Васильєв О. Програмування мовою PHP : навч. посіб. Збільшений формат В5. / Олексій Васильєв., 2022. – 368 с.
9. SQL [Електронний ресурс] - 2016. Режим доступу до ресурсу: <https://uk.wikibooks.org/wiki/SQL>
10. DeBarros A. Practical SQL: A Beginner's Guide to Storytelling with Data / Anthony DeBarros., 2018. – 392 с.
11. Shields W. SQL QuickStart Guide: The Simplified Beginner's Guide to Managing, Analyzing, and Manipulating Data With SQL (Coding & Programming - QuickStart Guides / Walter Shields., 2019. – 242 с.
12. Duckett J. PHP & MySQL: Server-side Web Development / Jon Duckett., 2022. – 672 с.
13. Документація бібліотеки JQuery [Електронний ресурс] // JQuery incorp. – 2016. – Режим доступу до ресурсу: <https://jquery.com>

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				55
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				56
Змн.	Арк.	№ докум.	Підпис	Дата		

Клас контролера Cart

```

<?php

namespace controllers;

use core\Controller;

class Cart extends Controller
{
    private $productModel;
    function __construct()
    {
        $this->productModel = new \models\Products();
    }
    /**
     * Перегляд кошику
     **/
    public function actionIndex()
    {
        if(isset($_SESSION['cart'])) {
            $whereStatement = implode(',', array_keys($_SESSION['cart']));
            $products = $this->productModel->getProductsWithWhereStatement('id IN (' .
$whereStatement . ')');
            for ($i = 0 ; $i < count($products); $i++)
                $products[$i]['quantity'] = $_SESSION['cart'][$products[$i]['id']];
        }
        else
            $products = null;
        return $this->render('index', ['productsInCart' => $products], [
            'MainTitle' => 'Кошик',
            'PageTitle' => 'Кошик'
        ]);
    }
}
/**

```

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				57
Змн.	Арк.	№ докум.	Підпис	Дата		

```

* Додавання до кошику
**/

public function actionAdd()
{
    if($this->isPost() && isset($_POST['product_id'], $_POST['quantity']))
    {
        $product_id = $_POST['product_id'];
        $quantity = $_POST['quantity'];
        $product = $this->productModel->getProductById($product_id);
        if ($product && $quantity > 0) {
            if (isset($_SESSION['cart']) && is_array($_SESSION['cart'])) {
                if (array_key_exists($product_id, $_SESSION['cart'])) {
                    $_SESSION['cart'][$product_id] += $quantity;
                } else {
                    $_SESSION['cart'][$product_id] = $quantity;
                }
            } else {
                $_SESSION['cart'] = array($product_id => $quantity);
            }
            header('Location: /cart/');
        }
    }
}

/**
* Оновлення кількості кошику
**/

public function actionUpdate()
{
    if (isset($_GET['id']) && isset($_GET['quantity']) && isset($_SESSION['cart'])) {
        $_SESSION['cart'][$_GET['id']] = $_GET['quantity'];
    }
    exit(json_encode(['quantity' => $_SESSION['cart'][$_GET['id']]));
}

/**
* Видалення з кошику

```

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				58
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    **/
    public function actionDelete()
    {
        if (isset($_GET['id']) && isset($_SESSION['cart']) &&
isset($_SESSION['cart'][$_GET['id']])) {
            unset($_SESSION['cart'][$_GET['id']]);
        }
        header('Location: /cart/');
    }
}

```

Додаток Б

Клас моделі Checkout

```

<?php

namespace models;

use core\Model;
use core\Utils;

class Checkout extends Model
{
    public function Validate($formRow)
    {
        $errors = [];
        if(empty($formRow['firstname']))
            $errors [] = 'Поле "Ім'я" не може бути порожнім';
        if(empty($formRow['lastname']))
            $errors [] = 'Поле "Прізвище" не може бути порожнім';
        if(empty($formRow['country']))
            $errors [] = 'Поле "Країна" не може бути порожнім';
        if(empty($formRow['address']))
            $errors [] = 'Поле "Адреса" не може бути порожнім';
        if(empty($formRow['city']))
            $errors [] = 'Поле "Місто" не може бути порожнім';
    }
}

```

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				59
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if(empty($formRow['phone']))
    $errors [] = 'Поле "Номер телефону" не може бути порожнім';
if(empty($formRow['email']))
    $errors [] = 'Поле "Пошта" не може бути порожнім';
if(count($errors) > 0)
    return $errors;
else
    return true;
}

public function AddCustomerDetails($customerRow)
{
    $validateResult = $this->Validate($customerRow);
    if(is_array($validateResult))
        return $validateResult;
    $fields = ['firstname', 'lastname', 'country', 'address', 'city', 'phone', 'email'];
    $customerRowFiltered = Utils::ArrayFilter($customerRow, $fields);
    $customerRowFiltered['date'] = date('Y-m-d H:i:s');
    \core\Core::getInstance()->getDB()->insert('customer_details', $customerRowFiltered);
    return true;
}

public function AddToOrder($idCustomer, $id, $quantity)
{
    \core\Core::getInstance()->getDB()->insert('orders', ['id_customer' => $idCustomer,
'id_product' => $id, 'quantity' => $quantity]);
    return true;
}

public function GetAllOrders($page_number, $limit_page)
{
    $offset = self::GetOffset($page_number, $limit_page);
    return \core\Core::getInstance()->getDB()->select('customer_details, products ,orders',
'customer_details.id_customer, firstname, lastname, phone, date, state, GROUP_CONCAT(name
SEPARATOR \' /\') as products_name, GROUP_CONCAT(quantity SEPARATOR \' /\') as

```

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				60
Змн.	Арк.	№ докум.	Підпис	Дата		

```

products_quantity', 'customer_details.id_customer = orders.id_customer AND products.id =
orders.id_product GROUP BY orders.id_customer, firstname, lastname, phone, date, state',
['date' => 'DESC'], $limit_page, $offset);
    }

    public function GetLastInsertedDetails()
    {
        $rows = \core\Core::getInstance()->getDB()->select('customer_details', '*', null,
['id_customer' => 'DESC'], 1);
        if (count($rows) > 0)
            return $rows[0];
        else
            return null;
    }

    public function UpdateState($state, $id_customer)
    {
        \core\Core::getInstance()->getDB()->update('orders', ['state' => $state], ['id_customer' =>
$id_customer]);
        return true;
    }

    public function GetTotalPages($page_count){
        $rows_count = self::GetCountOfOrders();
        return ceil((int)$rows_count / $page_count);
    }

    public function GetCountOfOrders()
    {
        $rows = \core\Core::getInstance()->getDB()->select('orders', 'COUNT(DISTINCT
id_customer)');
        if (count($rows) > 0)
            return $rows[0][0];
        else
            return null;
    }

```

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				61
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public function GetOffset($page_number, $page_count){
    if ($page_number == 1) {
        $offset = 0;
    } else {
        $offset = ($page_number - 1) * $page_count;
    }
    return $offset;
}
}

```

Додаток В

Клас контролера Checkout

```

<?php

namespace controllers;

use core\Controller;

class Checkout extends Controller
{
    protected $checkoutModel;
    protected $productModel;
    function __construct()
    {
        $this->checkoutModel = new \models\Checkout();
        $this->productModel = new \models\Products();
    }
    public function actionIndex()
    {
        $title = 'Оформлення замовлення';
        if(isset($_SESSION['cart'])) {
            $whereStatement = implode(',', array_keys($_SESSION['cart']));
            $products = $this->productModel->getProductsWithWhereStatement('id IN (' .
            $whereStatement . ')');

```

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				62
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$totalPrice = 0;
for ($i = 0 ; $i < count($products); $i++)
    $totalPrice += $products[$i]['price'] * $_SESSION['cart'][$products[$i]['id']];
}
else
    $totalPrice = 0;
if($this->isPost())
{
    $result = $this->checkoutModel->AddCustomerDetails($_POST);
    $idLastInsertedCustomer = $this->checkoutModel->GetLastInsertedDetails()['id_customer'];
    foreach ($_SESSION['cart'] as $id => $quantity){
        $this->checkoutModel->AddToOrder($idLastInsertedCustomer, $id, $quantity);
    }
    if($result === true) {
        unset($_SESSION['cart']);
        return $this->renderMessage('success', 'Ваше замовлення прийнято! Дякую за співпрацю!', null, [
            'PageTitle' => $title,
            'MainTitle' => $title
        ]);
    } else
    {
        $message = implode('<br/>', $result);
        return $this->render('index', ['totalPrice' => $totalPrice], [
            'PageTitle' => $title,
            'MainTitle' => $title,
            'MessageText' => $message,
            'MessageClass' => 'danger'
        ]);
    }
}
return $this->render('index', ['totalPrice' => $totalPrice], [
    'MainTitle' => $title,
    'PageTitle' => $title

```

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				63
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    });
}
}

```

Додаток Г

Клас контролера News

```

<?php

namespace controllers;

use core\Controller;

class News extends Controller
{
    protected $user;
    protected $newsModel;
    protected $userModel;
    public function __construct()
    {
        $this->userModel = new \models\Users();
        $this->newsModel = new \models\News();
        $this->user = $this->userModel->GetCurrentUser();
    }
    /**
     * Відображення новин
     */
    public function actionIndex()
    {
        $title = 'Новини';
        $lastNews = $this->newsModel->getNews();
        return $this->render('index', ['lastNews' => $lastNews], [
            'PageTitle' => $title,
            'MainTitle' => $title
        ]);
    }
}

```

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				64
Змн.	Арк.	№ докум.	Підпис	Дата		

```

/**
 * Перегляд обраної новини
 */
public function actionView()
{
    $id = $_GET['id'];
    $news = $this->newsModel->getNewsById($id);
    $title = $news['title'];
    return $this->render('view', ['model' => $news], [
        'PageTitle' => $title,
        'MainTitle' => $title
    ]);
}

/**
 * Додавання новини
 */
public function actionAdd()
{
    $titleForbidden = 'Доступ заборонено';
    if(empty($this->user) || $this->user['access'] != 1)
        return $this->renderMessage('error', 'Доступ до даної сторінки можуть мати
        тільки користувачі з відповідними правами доступу', null, [
            'PageTitle' => $titleForbidden,
            'MainTitle' => $titleForbidden
        ]);
    $title = 'Додавання новини';
    if($this->isPost())
    {
        $result = $this->newsModel->AddNews($_POST);
        if($result['error'] === false) {
            $allowedTypes = ['image/png', 'image/jpeg'];
            if(is_file($_FILES['photo']['tmp_name']) && in_array($_FILES['photo']['type'],
            $allowedTypes)) {
                switch ($_FILES['photo']['type'])
                {

```

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				65
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        case 'image/png':
            $extension = 'png';
            break;
        default:
            $extension = 'jpg';
    }
    $name = $result['id'].'_'.uniqid().'.'.$extension;
    move_uploaded_file($_FILES['photo']['tmp_name'], 'files/news/' . $name);
    $this->newsModel->ChangePhoto($result['id'], $name);
}
return $this->renderMessage('success', 'Новина успішно додана', null, [
    'PageTitle' => $title,
    'MainTitle' => $title
]);
}
else {
    $message = implode('<br/>', $result['messages']);
    return $this->render('form', ['model' => $_POST], [
        'PageTitle' => $title,
        'MainTitle' => $title,
        'MessageText' => $message,
        'MessageClass' => 'danger'
    ]);
}
}
return $this->render('form', ['model' => $_POST], [
    'PageTitle' => $title,
    'MainTitle' => $title
]);
}
/**
 * Редагування новини
 */
public function actionEdit()
{

```

		Коломієць А.О.			ІПЗ.КР.Б-121-24-ІПЗ	Арк.
		Фант М.О.				66
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $id = $_GET['id'];
        $news = $this->newsModel->getNewsById($id);
        $titleForbidden = 'Доступ заборонено';
        if(empty($this->user) || $this->user['access'] != 1)
            return $this->renderMessage('error', 'Доступ до даної сторінки можуть мати
тільки користувачі з відповідними правами доступу', null, [
                'PageTitle' => $titleForbidden,
                'MainTitle' => $titleForbidden
            ]);
        $title = 'Редагування новини';
        if($this->isPost())
        {
            $result = $this->newsModel->updateNews($_POST, $id);
            if($result === true) {
                $allowedTypes = ['image/png', 'image/jpeg'];
                if(is_file($_FILES['photo']['tmp_name']) && in_array($_FILES['photo']['type'],
$allowedTypes)) {
                    switch ($_FILES['photo']['type'])
                    {
                        case 'image/png':
                            $extension = 'png';
                            break;
                        default:
                            $extension = 'jpg';
                    }
                    $name = $id.'_'.uniqid().'.'.$extension;

                    move_uploaded_file($_FILES['photo']['tmp_name'], 'files/news/' . $name);
                    $this->newsModel->ChangePhoto($id, $name);
                }
                return $this->renderMessage('success', 'Новину успішно збережено', null, [
                    'PageTitle' => $title,
                    'MainTitle' => $title
                ]);
            } else

```

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				67
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        {
            $message = implode('<br/>', $result);
            return $this->render('form', ['model' => $news], [
                'PageTitle' => $title,
                'MainTitle' => $title,
                'MessageText' => $message,
                'MessageClass' => 'danger'
            ]);
        }
    }

    return $this->render('form', ['model' => $news], [
        'PageTitle' => $title,
        'MainTitle' => $title
    ]);
}

/**
 * Видалення новини
 */
public function actionDelete()
{
    $id = $_GET['id'];
    $titleForbidden = 'Доступ заборонено';
    if(empty($this->user) || $this->user['access'] != 1)
        return $this->renderMessage('error', 'Доступ до даної сторінки можуть мати
        тільки користувачі з відповідними правами доступу', null, [
            'PageTitle' => $titleForbidden,
            'MainTitle' => $titleForbidden
        ]);
    $title = 'Видалення новини';
    if(isset($_GET['confirm']) && $_GET['confirm'] == 'yes')
    {
        if($this->newsModel->deleteProducts($id))
            header('Location: /news/');
        else
            return $this->renderMessage('error', 'Помилка видалення новини', null, [

```

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				68
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'PageTitle' => $title,
        'MainTitle' => $title
    ]);
}
$news = $this->newsModel->getNewsById($id);
return $this->render('delete', ['model' => $news], [
    'PageTitle' => $title,
    'MainTitle' => $title
]);
}
}

```

Додаток Д

Клас моделі News

```

<?php

namespace models;

use core\Model;
use core\Utils;
use Imagick;

class News extends Model
{
    public function ChangePhoto($id, $file)
    {
        $folder = 'files/news/';
        $file_path = pathinfo($folder.$file);
        $file_big = $file_path['filename'].'_b';
        $file_medium = $file_path['filename'].'_m';
        $file_small = $file_path['filename'].'_s';
        $news = $this->getNewsById($id);
        if(is_file($folder.$news['photo'].'_b.jpg') && is_file($folder.$file))
            unlink($folder.$news['photo'].'_b.jpg');
        if(is_file($folder.$news['photo'].'_m.jpg') && is_file($folder.$file))

```

		Коломієць А.О.			ПІЗ.КР.Б-121-24-ПІЗ	Арк.
		Фант М.О.				69
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        unlink($folder.$news['photo'].'_m.jpg');
        if(is_file($folder.$news['photo'].'_s.jpg') && is_file($folder.$file))
            unlink($folder.$news['photo'].'_s.jpg');
        $news['photo'] = $file_path['filename'];
        $im_b = new Imagick();
        $im_b->readImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.$file);
        $im_b->cropThumbnailImage(1280, 1024, true);
        $im_b->
>writeImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.'/'.$file_big.'.jpg');
        $im_m = new Imagick();
        $im_m->readImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.$file);
        $im_m->cropThumbnailImage(270, 360, true);
        $im_m->
>writeImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.'/'.$file_medium.'.jpg');
        $im_s = new Imagick();
        $im_s->readImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.$file);
        $im_s->cropThumbnailImage(180, 240, true);
        $im_s->
>writeImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.'/'.$file_small.'.jpg');
        unlink($folder.$file);
        $this->updateNews($news, $id);
    }
    public function AddNews($newsRow)
    {
        $result = [];
        $validateResult = $this->Validate($newsRow);
        if (is_array($validateResult))
        {
            return [
                'error' => 'true',
                'messages' => $validateResult
            ];
        }
        $fields = ['title', 'text', 'short_text'];
        $newsRowFiltered = Utils::ArrayFilter($newsRow, $fields);

```

		Коломієць А.О.			ІІЗ.КР.Б-121-24-ІІЗ	Арк.
		Фант М.О.				70
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$newsRowFiltered['date'] = date('Y-m-d H:i:s');
$newsRowFiltered['photo'] = '...photo...';
$хid = \core\Core::getInstance()->getDB()->insert('news', $newsRowFiltered);
$result = [
    'error' => false,
    'id' => $хid
];
return $result;
}

public function getNews()
{
    return \core\Core::getInstance()->getDB()->select('news', '*', null, ['date' =>
'DESC']);
}

public function getLastNews($count)
{
    return \core\Core::getInstance()->getDB()->select('news', '*', null, ['date' =>
'DESC'], $count);
}

public function getNewsById($id)
{
    $news = \core\Core::getInstance()->getDB()->select('news', '*', ['id' => $id]);
    if (!empty($news))
        return $news[0];
    else
        return null;
}

public function updateNews($newsRow, $id)
{
    $validateResult = $this->Validate($newsRow);
    if (is_array($validateResult))

```

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				71
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return $validateResult;
$fields = ['title', 'text', 'short_text', 'photo'];
$newsRowFiltered = Utils::ArrayFilter($newsRow, $fields);
\core\Core::getInstance()->getDB()->update('news', $newsRowFiltered, ['id' =>
$id]);
return true;
}

public function deleteProducts($id)
{
    $news = $this->getNewsById($id);
    if(empty($news))
        return false;
    \core\Core::getInstance()->getDB()->delete('news', ['id' => $id]);
    return true;
}

public function Validate($row)
{
    $errors = [];
    if(empty($row['title']))
        $errors [] = 'Поле "Заголовок новини" не може бути порожнім';
    if(empty($row['short_text']))
        $errors [] = 'Поле "Короткий текст новини" не може бути порожнім';
    if(empty($row['text']))
        $errors [] = 'Поле "Повний текст новини" не може бути порожнім';
    if(count($errors) > 0)
        return $errors;
    else
        return true;
}
}

```

		Коломієць А.О.			ПЗ.КР.Б-121-24-ПЗ	Арк.
		Фант М.О.				72
Змн.	Арк.	№ докум.	Підпис	Дата		