

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри _____ Є.О. Давиденко
підпис

«__» _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Вебзастосунок магазину продажу квітів

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 408.1810802

Студент

_____ Ю.А. Андреева
підпис

«__» _____ 2022 р.

Керівник викладач кафедри ІПЗ, д-р філософії

_____ К.О. Антіпова
підпис

«__» _____ 2022 р.

Консультант канд. техн. наук, доцент

_____ А. О. Алексеева
підпис

«__» _____ 2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ

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

_____ Є.О. Давиденко

« ____ » _____ 2022 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи бакалавра

Видано студенту групи 408 факультету комп'ютерних наук

_____ Андреевій Юлії Андріївні _____.

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

«Веб застосунок магазину продажу квітів» _____.

Затверджена наказом по ЧНУ від «01» _____ грудня _____ 2022 р. № 314 _____.

2. Строк представлення кваліфікаційної роботи « » _____ 2022 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Вхідні дані до роботи – функціональні та нефункціональні вимоги до програмного забезпечення магазину продажу квітів. Результат – функціонуючий вебзастосунок магазину продажу квітів _____.

4. Перелік питань, що підлягають розробці

- аналіз методологій розробки програмного забезпечення, тобто дослідження предметної області;
- огляд існуючих вебзастосунків магазинів продажу квітів;

- розробка вимог до системи на основі інформації про предметну область та існуючі аналоги;
- розробка алгоритму вебзастосунку;
- проектування вебзастосунку магазину продажу квітів;
- програмна реалізація, тестування та відлагодження вебзастосунку для магазину продажу квітів.

5. Перелік графічних матеріалів:

Презентація _____.

6. Завдання до спеціальної частини

Дослідження питань охорони праці, які безпосередньо пов'язані з діяльністю розробника програмного забезпечення _____.

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Алексеева А. О.	Кафедра екології	Спеціальна частина з охорони праці

Керівник роботи викладач кафедри ІПЗ, д-р філософії Антіпова Катерина Олександрівна _____.

(посада, прізвище, ім'я, по батькові)

_____ (підпис)

Завдання прийнято до виконання

Андреева Юлія Андріївна

(прізвище, ім'я, по батькові студента)

_____ (підпис)

Дата видачі завдання « _____ » _____ 2022 р.

АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Вебзастосунок магазину продажу квітів»

Студент 408 гр.: Андрєєва Юлія Андріївна

Керівник: викладач кафедри ІІЗ, д-р філософії Антіпова К. О.

Об'єкт кваліфікаційної роботи – процес конструювання Інтернет-магазину букетів квітів.

Предмет кваліфікаційної роботи – програмні засоби та методи проектування програмного забезпечення з використанням ASP .NET MVC.

Мета кваліфікаційної роботи – вдосконалення інтернет-магазину продажу квітів за рахунок зменшення кількості модальних вікон з рекламою та акціями.

У першому розділі представлено аналіз вимог до системи, до якого входить опис предметної області, аналіз аналогічних систем, постановка задачі та сценарії використання системи. У другому розділі здійснено проектування та моделювання системи. У третьому розділі показано основні кроки реалізації застосунку. В останньому розділі розглянуто питання охорони праці, які безпосередньо пов'язані з діяльністю розробника програмного забезпечення.

В результаті виконання кваліфікаційної роботи бакалавра було реалізовано вебзастосунок магазину продажу квітів. Також були зроблені висновки щодо вдосконалення інтернет-магазину продажу квітів.

КРБ викладена на 47 сторінки, вона містить 4 розділи, 29 ілюстрацій, 18 таблиці, 18 джерел в переліку посилань

Ключові слова: ASP .NET, вебзастосунок, інтернет магазин, база даних, MS SQL, MVC.

ABSTRACT

of the Bachelor's Thesis

"Flower shop web application"

Student of group 408: Andreieva Yuliia Andriyivna

Supervisor: Lecturer of the Department of SE, Dr. of Philosophy Antipova K. O

The object of qualification work is the process of designing an online store of flower bouquets.

Subject of qualification work - software and methods of software design using ASP .NET MVC.

The purpose of the qualification work is to improve the online flower shop by reducing the number of modal windows with advertising and promotions.

The first section presents an analysis of the requirements for the system, which includes a description of the subject area, analysis of similar systems, problem statement and scenarios for using the system. In the second section, the system is designed and modeled. The third section shows the main steps of application implementation. The last section discusses health and safety issues that are directly related to the activities of the software developer.

As a result of the bachelor's qualification work, a web application of a flower shop was implemented. Conclusions were also made on improving the online flower shop.

The CRB is set out on 47 pages, it contains 4 sections, 29 illustrations, 18 tables, 18 sources in the list of links

Keywords: ASP .NET, web application, online store. database, MS SQL, MVC.

ЗМІСТ

ВСТУП.....	3
1 АНАЛІЗ ВИМОГ ДО СИСТЕМИ.....	5
1.1 Опис предметної області	5
1.2 Аналіз аналогічних систем.....	6
1.3 Постановка задачі.....	11
Висновки до розділу 1	12
2 ПРОЄКТУВАННЯ ТА МОДЕЛЮВАННЯ СИСТЕМИ.....	14
2.1 Варіанти використання (Use cases)	14
2.2 Сценарії використання.....	16
2.3 Діаграма класів (Class diagram)	24
2.4 Діаграма станів (State Diagram)	27
2.5 Проєктування бази даних	28
2.6 Діаграма пакетів (Package diagram).....	30
Висновки до розділу 2	31
3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	32
3.1 Обґрунтування вибору технологій для реалізації системи.....	32
3.2 Опис програмної реалізації	35
3.3 Керівництво користувача	40
Висновки до розділу 3	47
ВИСНОВКИ.....	48
ПЕРКЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	49
ДОДАТОК А BOUQUETS CONTROLLER.....	51
ДОДАТОК Б SHOPPING CART CONTROLLER.....	53
ДОДАТОК В USERS ADMIN CONTROLLER	55
ДОДАТОК Г ROLE VIEW MODEL	59

ВСТУП

Програмне забезпечення для Інтернет-магазину букетів сприяє підвищенню ефективності роботи відповідного магазину за рахунок представлення асортименту у вигляді Інтернет-вітрини та автоматизації обліку даних про замовлення. Крім того, робота над курсовим проектом допомагає покращити знання в області конструювання програмного забезпечення.

Актуальність теми зумовлена тим, що вміння проектувати Інтернет-магазини є популярним на ринку праці у зв'язку з більшим переходом на цифрові технології. Оскільки всі Інтернет-магазини мають аналогічний функціонал, то на базі створених в ході виконання роботи матеріалів можна проектувати Інтернет-магазини іншої тематики та рівня складності. Даний інтернет-магазин покращений відсутністю великої кількості реклами та акцій, тому інтерфейс не є перенавантаженим.

Темою даної КРБ є вебзастосунок магазину продажу квітів.

Об'єктом КРБ є процес конструювання Інтернет-магазину букетів квітів.

Предметом КРБ є програмні засоби та методи проектування програмного забезпечення з використанням ASP .NET MVC.

Метою КРБ є вдосконалення інтерфейсу інтернет-магазину продажу квітів за рахунок зменшення кількості модальних вікон з рекламою та акціями.

Для досягнення поставленої мети необхідно виконати **наступні завдання**:

- 1) Аналіз аналогічних систем;
- 2) Створити технічне завдання (ТЗ);
- 3) Провести аналіз вимог до системи;
- 4) Провести аналіз системи;
- 5) Здійснити проектування системи;
- 6) Розробити інтернет магазин.

Для розв'язання поставлених завдань буде використана мова програмування С# (ASP NET MVC).

Актуальність спеціального розділу про охорону праці зумовлена тим, що розповсюдження та широке використання інформаційних технологій сприяло розробці норм та вимог до використання комп'ютерної техніки на підприємствах. Дані норми та вимоги регулюються законодавством України. Саме тому дуже важливо створювати належні умови праці, виключати або мінімізувати ризики для життя та здоров'я робітників.

Метою спеціального розділу про охорону праці є дослідження питань охорони праці, які безпосередньо пов'язані з діяльністю розробника програмного забезпечення.

1 АНАЛІЗ ВИМОГ ДО СИСТЕМИ

1.1 Опис предметної області

Інтернет-магазин – спеціальний вебсайт, який містить певний каталог, за допомогою якого відбувається продаж товарів, включаючи їх доставку покупцеві. Відмінністю такого магазину є відсутність фізичної особи, що є перевагою під час пандемії, оскільки немає ризику захворіти. Такий інтернет-магазин має ряд інших переваг, а саме:

1. Охоплює більшу аудиторію покупців;
2. Є можливість переглянути всю характеристику будь-якого товару;
3. Порівняння товарів та цін;
4. Можливість переглянути товар у будь-якому місці та у будь-який час;
5. Економія часу та коштів, оскільки не потрібно їх витратити на дорогу;
6. В умовах воєнного часу це безпечно;
7. Значно менші витрати на утримання та організацію роботи, оскільки у ньому значно обмеженіша матеріально-технічна база.

Але у будь-якого інтернет-магазину є і недолік – неможливо визначити чи відповідає товар заявленим параметрам якості. А також можливе шахрайство при здійсненні грошових транзакцій та можливі проблеми із доставкою.

Кількість представлених на сервері видів товарів може коливатися від кількох одиниць до кількох тисяч.

Основними вимогами до інтернет-магазинів є:

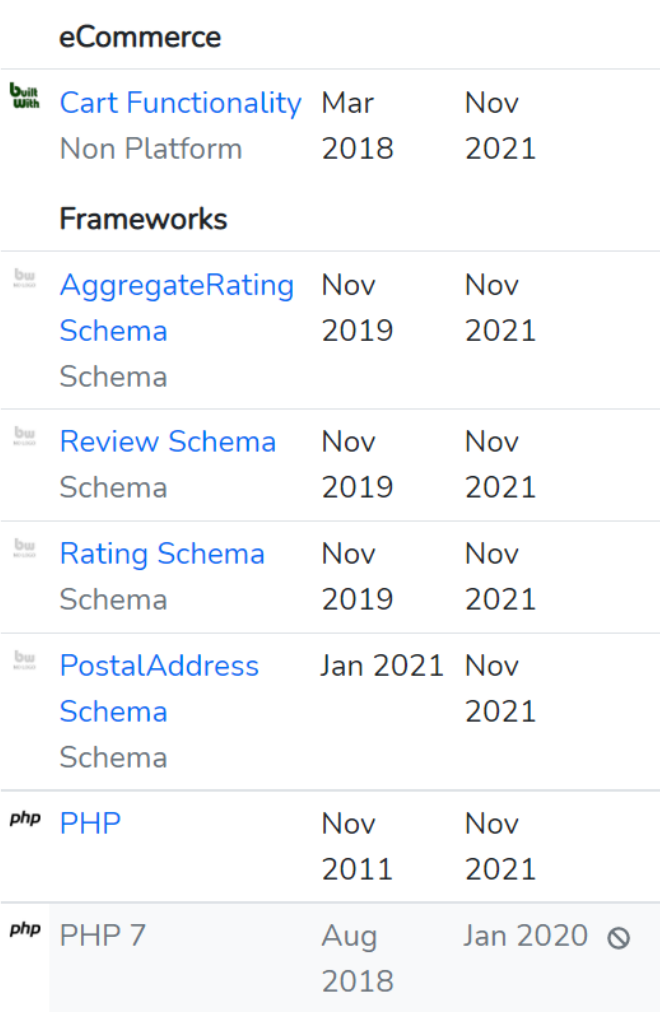
1. Простий та зрозумілий інтерфейс;
2. Відсутність великої кількості реклами та акцій;
3. Зручна система посилань, що дозволяє оптимальним способом одержати необхідну користувачеві інформацію;
4. Мінімальна кількість дій користувача для здійснення покупки.

1.2 Аналіз аналогічних систем

Дані системи створені для бізнесу і не є open-source проектами, тому отримати детальну інформацію про процес розробки, виробника та архітектуру проблематично. Для визначення того, які саме технології (мови реалізації) використані при розробці даних сайтів використано спеціалізований ресурс [6]. Аналіз схожих систем представлено в таблицях 1, 2 та 3.

Flowers

Таблиця 1.1 – Характеристики автоматизованої системи «FLOWERS»

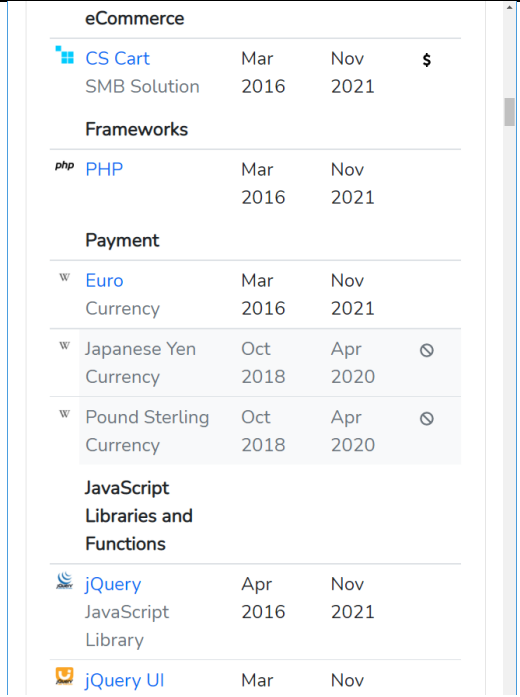
Характеристика	Значення характеристики
Посилання	https://flowers.ua/ru/nikolaev
Мови програмування: <ul style="list-style-type: none"> – PHP – Angular – JavaScript Детальний список фреймворків та бібліотек на Рис.1.	 <p>The screenshot displays a list of technologies used in the Flowers system, categorized into eCommerce, Frameworks, and PHP. The eCommerce section includes 'Cart Functionality' (Mar 2018 - Nov 2021) and 'Non Platform'. The Frameworks section lists 'AggregateRating Schema' (Nov 2019 - Nov 2021), 'Review Schema' (Nov 2019 - Nov 2021), 'Rating Schema' (Nov 2019 - Nov 2021), and 'PostalAddress Schema' (Jan 2021 - Nov 2021). The PHP section includes 'PHP' (Nov 2011 - Nov 2021) and 'PHP 7' (Aug 2018 - Jan 2020).</p>
	<p>Рисунок 1 – Технології для автоматизованої системи «FLOWERS»</p>

Продовження таблиці 1.1

Характеристика	Значення характеристики
Функції	<ol style="list-style-type: none"> 1. Пошук інформації про товари 2. Перегляд інформації про товари 3. Занесення товарів у кошик 4. Здійснення замовлення 5. Написання відгуків та питань 6. Користування чатом на сайті 7. Авторизація 8. Реєстрація

FLORA24

Таблиця 1.2 – Характеристики автоматизованої системи «FLORA24»

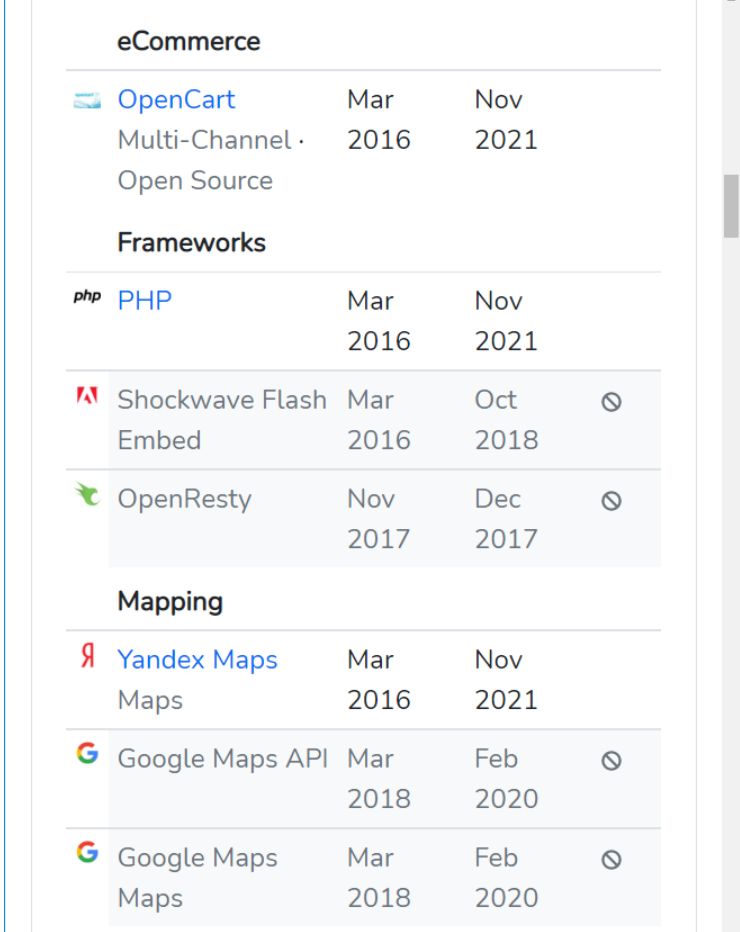
Характеристика	Значення характеристики
Посилання	http://nikolaev.flora24.com.ua/
Мови програмування: – PHP Детальний список фреймворків та бібліотек на Рис.2.	 <p>The screenshot displays a list of technologies categorized into several groups: eCommerce (CS Cart, SMB Solution), Frameworks (PHP), Payment (Euro Currency, Japanese Yen Currency, Pound Sterling Currency), and JavaScript Libraries and Functions (jQuery, jQuery UI). Each entry includes a logo, the technology name, and associated dates (e.g., Mar 2016, Nov 2021).</p>
	<p>Рисунок 2 – Технології для автоматизованої системи «FLORA24»</p>

Продовження таблиці 1.2

Характеристика	Значення характеристики
<i>Функції</i>	<ol style="list-style-type: none">1. Пошук інформації про товари2. Перегляд інформації про товари3. Порівняння товарів4. Занесення товарів у відкладені5. Занесення товарів у кошик6. Здійснення замовлення7. Написання відгуків та питань8. Користування чатом на сайті9. Авторизація10. Реєстрація

Тэрика галерея цветов

Таблиця 1.3 – Характеристики автоматизованої системи «Тэрика галерея ЦВЕТОВ»

Характеристика	Значення характеристики																																									
Посилання	https://galleryflowers.com.ua/																																									
<p>Мови програмування:</p> <p>– PHP</p> <p>Детальний список фреймворків та бібліотек на Рис.3.</p>	 <p>The screenshot shows a list of technologies used in the system, categorized into eCommerce, Frameworks, and Mapping. Each entry includes a logo, the technology name, and its release dates.</p> <table border="1"> <thead> <tr> <th>Category</th> <th>Technology</th> <th>Release Date 1</th> <th>Release Date 2</th> <th>Additional Info</th> </tr> </thead> <tbody> <tr> <td rowspan="3">eCommerce</td> <td>OpenCart Multi-Channel · Open Source</td> <td>Mar 2016</td> <td>Nov 2021</td> <td></td> </tr> <tr> <td colspan="4">Frameworks</td> </tr> <tr> <td>PHP</td> <td>Mar 2016</td> <td>Nov 2021</td> <td></td> </tr> <tr> <td rowspan="3">Mapping</td> <td>Shockwave Flash Embed</td> <td>Mar 2016</td> <td>Oct 2018</td> <td>⊗</td> </tr> <tr> <td>OpenResty</td> <td>Nov 2017</td> <td>Dec 2017</td> <td>⊗</td> </tr> <tr> <td>Yandex Maps</td> <td>Mar 2016</td> <td>Nov 2021</td> <td></td> </tr> <tr> <td></td> <td>Google Maps API</td> <td>Mar 2018</td> <td>Feb 2020</td> <td>⊗</td> </tr> <tr> <td></td> <td>Google Maps Maps</td> <td>Mar 2018</td> <td>Feb 2020</td> <td>⊗</td> </tr> </tbody> </table> <p>Рисунок 3 – Технології для автоматизованої системи «Тэрика галерея цветов»</p>	Category	Technology	Release Date 1	Release Date 2	Additional Info	eCommerce	OpenCart Multi-Channel · Open Source	Mar 2016	Nov 2021		Frameworks				PHP	Mar 2016	Nov 2021		Mapping	Shockwave Flash Embed	Mar 2016	Oct 2018	⊗	OpenResty	Nov 2017	Dec 2017	⊗	Yandex Maps	Mar 2016	Nov 2021			Google Maps API	Mar 2018	Feb 2020	⊗		Google Maps Maps	Mar 2018	Feb 2020	⊗
Category	Technology	Release Date 1	Release Date 2	Additional Info																																						
eCommerce	OpenCart Multi-Channel · Open Source	Mar 2016	Nov 2021																																							
	Frameworks																																									
	PHP	Mar 2016	Nov 2021																																							
Mapping	Shockwave Flash Embed	Mar 2016	Oct 2018	⊗																																						
	OpenResty	Nov 2017	Dec 2017	⊗																																						
	Yandex Maps	Mar 2016	Nov 2021																																							
	Google Maps API	Mar 2018	Feb 2020	⊗																																						
	Google Maps Maps	Mar 2018	Feb 2020	⊗																																						
Функції	<ol style="list-style-type: none"> 1. Пошук інформації про товари 2. Перегляд інформації про товари 3. Порівняння товарів 4. Занесення товарів у список обраного 																																									

Продовження таблиці 1.3

Характеристика	Значення характеристики
<i>Функції</i>	5. Занесення товарів у кошик 6. Здійснення замовлення 7. Написання відгуків та питань 8. Перегляд рекомендованих товарів до вказаного товару 9. Користування чатом на сайті 10. Авторизація 11. Реєстрація

Висновки про переваги та недоліки даних систем можна робити лише з точки зору досвіду користувача. У відкритому доступі немає даних про кількість відмов, баги, швидкість обробки запиту і т.д.

Інтерфейс та функціонал цих систем майже однаковий, немає відчутної різниці у швидкості роботи.

Висновки про переваги та недоліки подані у таблиці 4.

Таблиця 1.4 – Переваги та недоліки аналогічних сайтів

Критерії	FLOWERS	FLORA24	Тэрика галерея цветов
Перегляд рекомендованих товарів до вказаного товару	–	–	+
Занесення товарів у список обраного	–	+	+

Продовження таблиці 1.4

Критерії	FLOWERS	FLORA24	Тэрика галерея цветов
Порівняння товарів	–	+	+
Перевантаженість інтерфейсу	+	+	+
Велика кількість модальних вікон з рекламою та акціями	+	+	+

1.3 Постановка задачі

Опис основних функцій системи

Програмне забезпечення автоматизованого магазину продажу квітів, яке розробляється в ході бакалаврської дипломної роботи, має наступний набір функціональних можливостей:

1. Пошук інформації про товари;
2. Перегляд інформації про товари;
3. Створення, редагування та видалення товарів;
4. Порівняння товарів;
5. Занесення товарів у список обраного;
6. Занесення товарів у кошик;
7. Здійснення замовлення;
8. Рекомендації щодо підбору аксесуарів до товару;
9. Написання відгуків;
10. Користування чатом на сайті;
11. Двухфакторна авторизація;

12. Реєстрація та авторизація користувачів в системі (з використанням Google) з наступним розмежуванням прав доступу в залежності від ролі.

Опис ролей користувачів в системі

- Гість (неавторизований користувач);
- Зареєстрований користувач;
- Адміністратор;
- Менеджер;
- Розробник.

Опис структури системи

Програмне забезпечення, що розробляється, має наступні складові:

- Backend (серверна частина вебзастосунку, відповідає за доступ до даних).
- Frontend (клієнтська частина вебзастосунку, відповідає за представлення).
- База даних.

Висновки до розділу 1

У результаті написання розділу 1 КРБ було виконано завдання для досягнення мети, яка полягала в аналізі предметної області та вимог до системи, що розробляється.

Для досягнення поставленої мети необхідно виконати *наступні завдання*:

- 1) Аналіз аналогічних систем;
- 2) Створити технічне завдання (ТЗ);
- 3) Провести аналіз вимог до системи;
- 4) Провести аналіз системи;
- 5) Здійснити проектування системи;
- 6) Розробити інтернет магазин.

У розділі 1 було проведено аналіз предметної області. Також було проаналізовано наявні аналоги. На основі проведеного аналізу було з'ясовано, що інтернет магазини широко використовуються. Кожен розглянутий аналог має свої переваги та недоліки. Актуальність розробки власного інтернет магазину полягає також у викоріненні таких мінусів: перевантаженість інтерфейсу та велика кількість модальних вікон з рекламою та акціями. Також було поставлено завдання, у якому були описані основні функції системи, ролі користувачів у системі та описана структура системи.

2 ПРОЄКТУВАННЯ ТА МОДЕЛЮВАННЯ СИСТЕМИ

2.1 Варіанти використання (Use cases)

Варіант використання фіксує згоду між користувачами системи про її поведінку. Варіант використання описує поведінку системи при її відповідях на запити дійових осіб (actors) в різних умовах. Діаграма варіантів використання [16], зображена на рисунку 2.1, допомагає систематизувати функціональні вимоги та визначити ролі користувачів разом з відповідними правами доступу. Представлено ті типи зв'язків, що мають місце у застосунку, тобто *generalization*, *extend* та *include relationship*.

Інформацію про акторів представлено у таблиці 2.1.

Таблиця 2.1 – Інформація про акторів

Назва	Опис
Гість	Незарєєстрований користувач, який має доступ лише до перегляду контенту сайту.
Зарєєстрований користувач	Має можливість здійснювати та відстежувати свої замовлення
Менеджер	Керує даними про замовлення та формує звіти про продажі.
Адміністратор	Наповнює сайт інформацією, редагує, видаляє та додає інформацію.

Кафедра інженерії програмного забезпечення
 Веб застосунок магазину продажу квітів

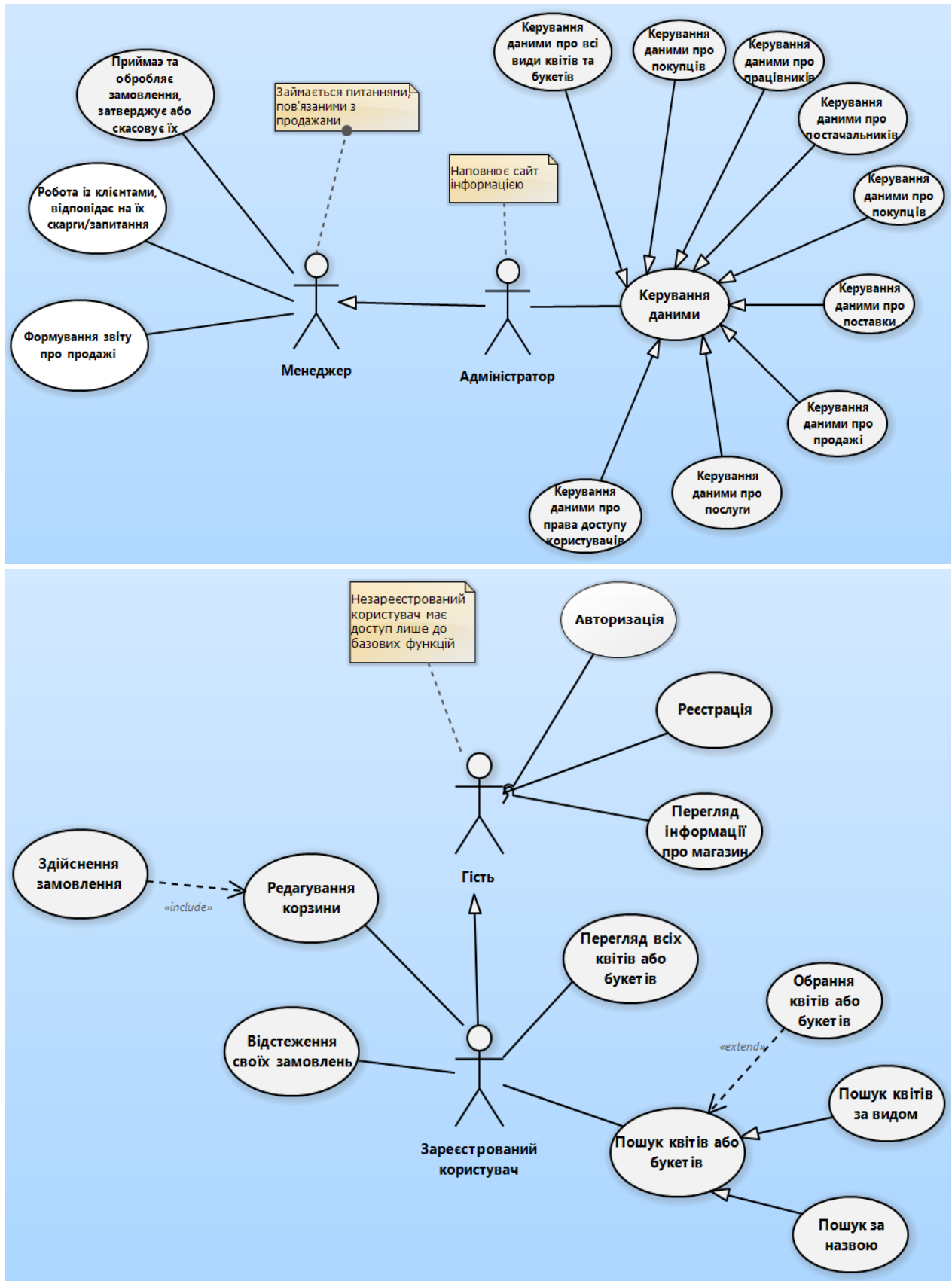


Рисунок 2.1 – Діаграма варіантів використання системи
 На діаграмі відображено всі функції системи.

2.2 Сценарії використання

Сценарії використання дозволяють сформулювати та задокументувати вимоги для полегшення розробки та підтримки застосунку [17]. Для демонстрації можливих форм написання сценаріїв, перші 3 сценарії представлені в:

1. Короткій формі;
2. Поверхневій формі;
3. Повній формі.

Сценарій №1 (коротка форма): Пошук товарів

Користувач переходить до головної сторінки сайту. Вводить у спеціалізоване поле назву товару або обирає категорію товарів. Система здійснює пошук в БД за поданими характеристиками. Користувач в отриманому списку результатів переходить до сторінки з потрібним товаром.

Сценарій №2 (поверхнева форма): Здійснення замовлення

Головний сценарій:

- Користувач додає товари в кошик.
- Користувач редагує набір товарів в кошику (кількість, наявність).
- Користувач натискає кнопку для переходу на сторінку здійснення замовлення.
- Користувач заповнює контактні дані у відповідній формі.
- Система здійснює валідацію контактних даних.
- Користувач отримує доступ та автоматично переходить до форми вибору способу оплати, введення промокоду та доставки.
- Користувач заповнює дані про спосіб оплати, промокод та доставку.
- Система здійснює валідацію даних про спосіб оплати та доставку.
- Користувач обирає пакування.
- Користувач відправляє замовлення.

Альтернативні сценарії:

- 1) Технічний збій роботи системи, видається відповідне повідомлення.
- 2) Користувач змінив рішення щодо покупки, тому вимагає скасувати замовлення (звернення до служби підтримки).
- 3) Сталася помилка при розрахунку кількості товарів на складі, тому менеджер має вручну змінити склад та ціну замовлення, зателефонувавши покупцю.

Таблиця 2.2 – Сценарій №3 (повний): Авторизація в системі

Primary Actor	Покупець
Scope	Інтернет-магазин букетів
Level	Sub-function
Preconditions	Покупець не авторизований.
Stakeholders and interests	<ol style="list-style-type: none"> 1. Покупець – авторизація полегшує: <ul style="list-style-type: none"> – Процес замовлення (не треба кожного разу заново вводити контактні дані). – Процес відстежування замовлення. – Накопичення знижок та персоналізованих рекомендацій на основі попередніх замовлень 2. Компанія (продавець) – полегшення збору даних для аналітики, формування рекомендацій та спілкування з клієнтом.
Main Success Scenario:	
<ol style="list-style-type: none"> 1. Покупець вводить номер телефону. 2. Покупець вводить пароль. 3. Система здійснює валідацію даних. 4. Покупець авторизувався в системі. 	
Result	Покупець авторизувався в системі.

Продовження таблиці 2.2

Extensions	
*a	Система не може підключитися до БД. Система виводить відповідне повідомлення. Result: користувач не авторизований.
1a	Обрано авторизацію через обліковий запис Google.
2a	Обрано авторизацію через Facebook.
3a	Перевищено допустиму кількість спроб авторизації. Система виводить відповідне повідомлення. Result: користувач не авторизований.
4a	Не знайдено користувача з таким номером телефону. Система виводить відповідне повідомлення. Result: користувач не авторизований.
Special Requirements:	Адаптивність інтерфейсу
Technology and Data Variations List:	<ol style="list-style-type: none"> 1) Запам'ятовувати користувача при авторизації, щоб кожного разу не вводити логін та пароль. 2) Можливість обрати мову (англійська, українська, російська). 3) Допустима кількість спроб авторизації – 5 спроб.
Frequency of Occurrence	Система може працювати майже безперервно.
Miscellaneous (Open Issues)	Забезпечити можливість відновлення доступу до облікового запису у випадку, якщо користувач не може згадати пароль від облікового запису.

Таблиця 2.3 – Сценарій №4: Кастомізувати сайт

Діючі особи	Покупець, система
Мета	Обрати мову та місто
Успішний сценарій:	
<ol style="list-style-type: none"> 1) Покупець обрав місто та мову. 2) Система виводить інформацію обраною мовою та відфільтровує інформацію про магазини та пункти доставки відповідно до міста. 	
Результат	Збережено обрані налаштування.

Таблиця 2.4 – Сценарій №5: Переглянути інформацію про магазин

Діючі особи	Покупець, система
Мета	Отримати дані про товари, способи оплати та доставки.
Успішний сценарій:	
<ol style="list-style-type: none"> 1) Покупець обрав розділ про нас. 2) Система виводить відповідну інформацію. 3) Покупець здійснює пошук серед отриманих записів за категоріями або назвою. 4) Система виводить необхідні записи. 	
Результат	Покупець отримав способи доставки/оплати.

Таблиця 2.5 – Сценарій №6: Перегляд товару

Діючі особи	Покупець, система
Мета	Переглянути всю доступну інформацію про певний товар.
Успішний сценарій:	
<ol style="list-style-type: none"> 1) Покупець переходить до сторінки з товаром. 2) Обирає необхідні параметри товару (вид та назву). 3) Система зберігає ці налаштування. 4) Покупець дивиться інформацію в певних розділах (головне, характеристики, опис, відгуки та питання, аксесуари, наявність в магазинах). 	
Результат	Обрано товар.

Таблиця 2.6 – Сценарій №7: Залишити відгук

Діючі особи	Покупець, система.
Мета	Залишити відгук на певний товар.
Успішний сценарій:	
<ol style="list-style-type: none"> 1) Покупець заповнює контактну інформацію. 2) Пише текст відгука. 3) Ставить оцінку. 4) Обирає збереження відгуку. 5) Система записує відгук до БД. 	
Результат	Відгук збережено.

Таблиця 2.7 – Сценарій №8: Порівняти товари

Діючі особи	Покупець, система
Мета	Отримати порівняльну характеристику товарів.
Передумова	До порівняння додано хоча б два товари.
Успішний сценарій:	
<ol style="list-style-type: none"> 1) Покупець переходить на сторінку з порівнянням товарів. 2) Обирає характеристики для порівняння. 3) За потреби видаляє або переносить до кошика певний товар. 	
Результат	Покупець отримує порівняльну характеристику товарів.

Таблиця 2.8 – Сценарій №9: Переглянути статус замовлення

Діючі особи	Покупець, система.
Мета	Отримати інформацію про статус замовлення.
Успішний сценарій:	
<ol style="list-style-type: none"> 1) Перейти до відповідного розділу сайту. 2) Ввести номер замовлення. 3) Система здійснює пошук даних про замовлення в БД. 4) Система виводить повідомлення про статус замовлення. 	
Результат	Покупець отримує інформацію про статус замовлення.

Таблиця 2.9 – Сценарій №10: Звернутися до менеджера

Діючі особи	Покупець, менеджер, система
Мета	Отримати інформацію від менеджера.
Успішний сценарій:	
<ol style="list-style-type: none"> 1) Покупець обирає спосіб зв'язку (чат, телефонний дзвінок або питання на сторінці з товаром). 2) Покупець пише свої контактні дані і текст повідомлення. 3) Система направляє повідомлення менеджеру. 4) Менеджер отримує повідомлення. 5) Покупець отримує відповідь від менеджера. 	
Результат	Покупець отримує відповідь на своє питання від менеджера.

Таблиця 2.10 – Сценарій №11: Реєстрація

Діючі особи	Покупець, система
Мета	Створити обліковий запис.
Передумова	Покупець не авторизований.
Успішний сценарій:	
<ol style="list-style-type: none"> 1. Покупець переходить до форми реєстрації. 2. Заповнює контактні дані. 3. Система здійснює валідацію даних. 4. Система зберігає обліковий запис. 5. Покупець отримує повідомлення про успішну реєстрацію. 	
Результат	Збережено новий обліковий запис.

Таблиця 2.11 – Сценарій №12: Додати товар в список бажань

Діючі особи	Покупець, система
Мета	Зберегти обраний товар.
Передумова	Покупець авторизований.
Успішний сценарій:	
<ol style="list-style-type: none"> 1. Покупець обирає товар. 2. Покупець додає товар в список бажань 3. Система зберігає дані про список бажань даного покупця. 4. Система оновлює інформацію про те скільки людей додали товар в список бажань на сторінці з цим товар. 	
Результат	Обрані товари збережено у списку бажань певного покупця.

Таблиця 2.12 – Сценарій №13: Налаштування та перегляд облікового запису

Діючі особи	Покупець, система
Мета	Налаштувати обліковий запис.
Передумова	Покупець авторизований.
Успішний сценарій:	
<ol style="list-style-type: none"> 1. Покупець переходить до облікового запису. 2. Переходить до необхідного розділу. 3. Редагує дані. 4. Зберігає дані. 5. Система зберігає зміни в БД. 	
Результат	Збережено налаштування облікового запису.

2.3 Діаграма класів (Class diagram)

Діаграма класів показує структуру класів застосунка, зв'язки між ними та атрибути кожного окремого класу [3]. Вона є частиною уніфікованої мови моделювання (UML).

У свою чергу UML — це мова графічного опису для моделювання об'єктів у розробці програмного забезпечення, бізнес-процесів, системної інженерії та візуалізації організаційної структури. У своєму описі вона містить:

- класи;
- їх атрибути;
- операції (або методи);
- взаємозв'язки між об'єктами.

На рис. 2.2 зображена діаграма класів до інтернет магазину букетів квітів, на якій представлені лише класи. Контролери та представлення не відображені, оскільки буде дотримано стандарт при іменуванні класів.

Кафедра інженерії програмного забезпечення
Веб застосунок магазину продажу квітів

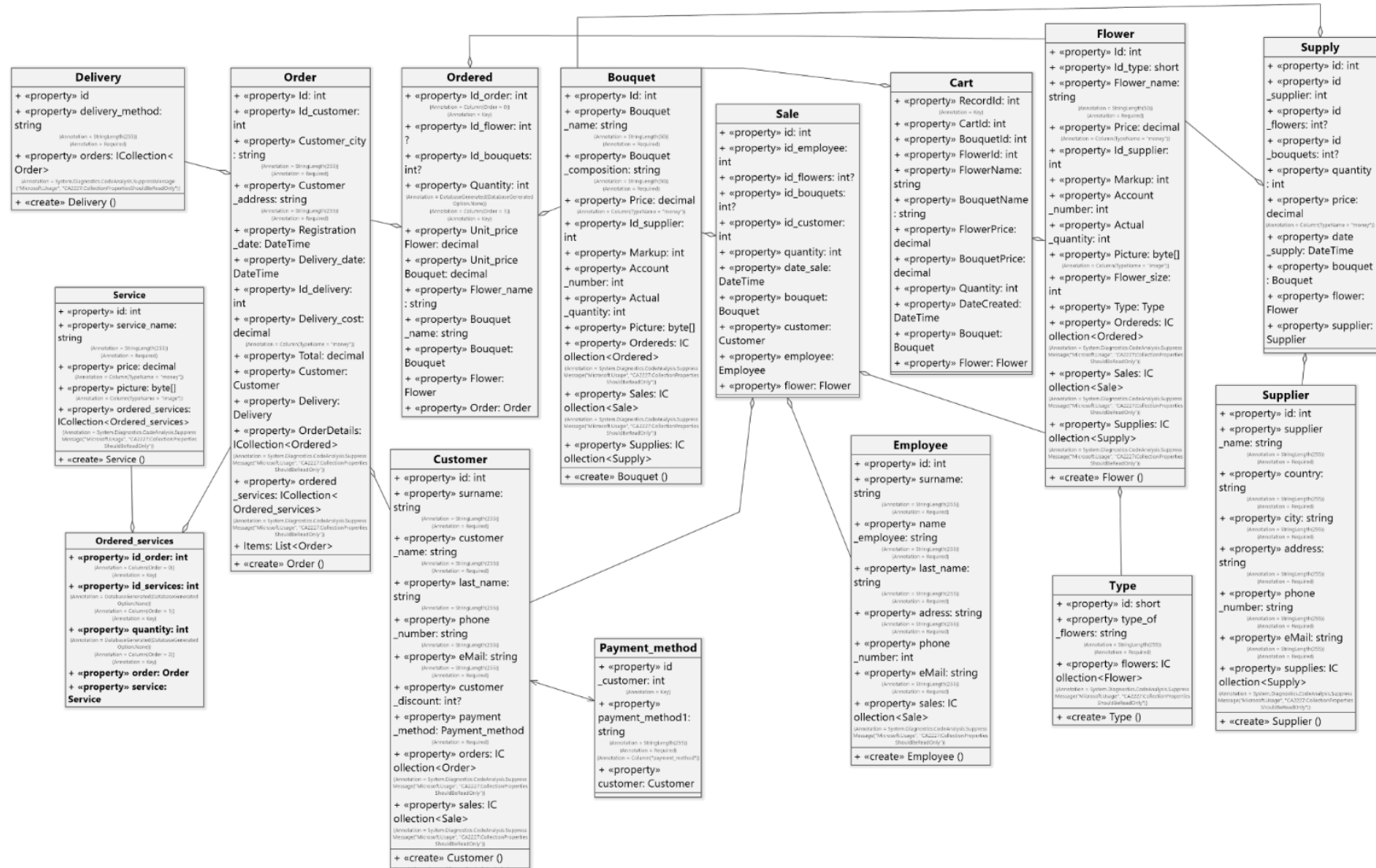


Рисунок 2.2 – Діаграма класів

2.4 Діаграма станів (State Diagram)

Діаграма станів визначає зміну станів об'єкту у часі (одна з діаграм моделювання поведінки в UML) [14]. На рисунку 2.3 зображено діаграму станів для всього застосунку.

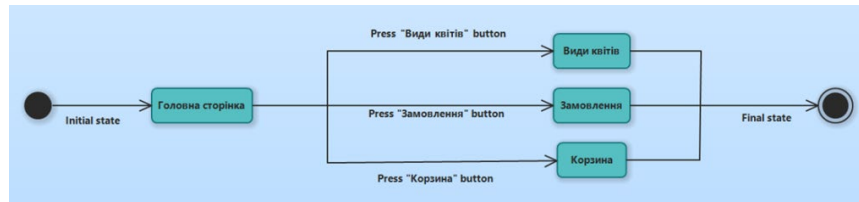


Рисунок 2.3 – Діаграма станів для програмного забезпечення Інтернет-магазину букетів

На рисунку 2.4 зображено діаграму станів для процесу здійснення замовлення у системі.

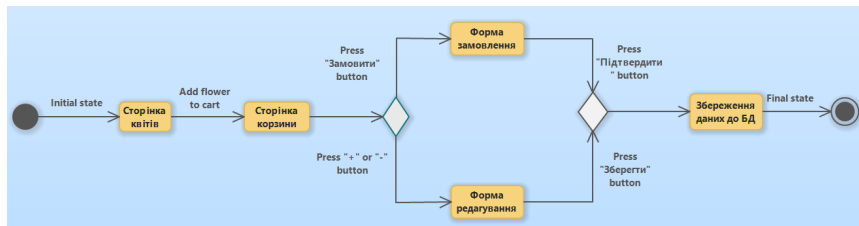


Рисунок 2.4 – Діаграма станів для здійснення замовлення

На рисунку 2.5 зображено діаграму станів для процесу здійснення CRUD-операцій над даними про квітку.

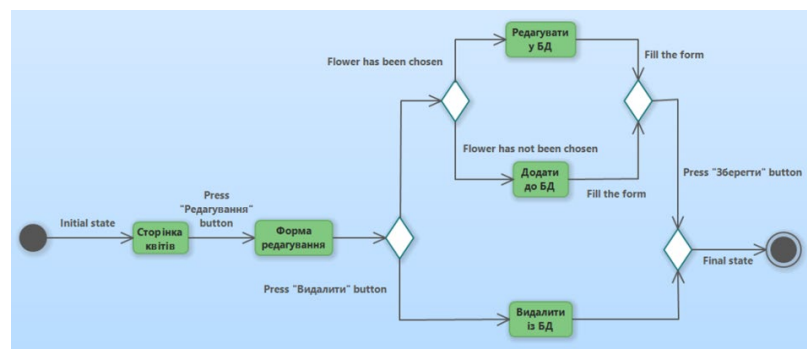


Рисунок 2.5 – Діаграма станів для здійснення CRUD-операцій над даними про квітку

2.5 Проєктування бази даних

База даних має перебувати в 3NF. Для цього вона має спочатку перебувати в 2NF та 1NF. Для перебування в 1NF треба виконати наступні вимоги:

1. Кожна колонка має містити атомарне значення.
2. Кожна колонка має містити значення одного типу.
3. Кожна колонка мусить мати унікальне ім'я.
4. Порядок зберігання даних не має значення.

Для приведення у 2NF треба виключити залежність атрибутів від частини складеного первинного ключа (виключити Partial Dependencies). Для приведення в 3NF необхідно виключити залежність одного неключового атрибута від іншого неключового атрибута (Transitive Dependency). Отже, нормалізувавши БД, маємо схему даних, зображену на рисунку 2.6.

Кафедра інженерії програмного забезпечення
 Веб застосунок магазину продажу квітів

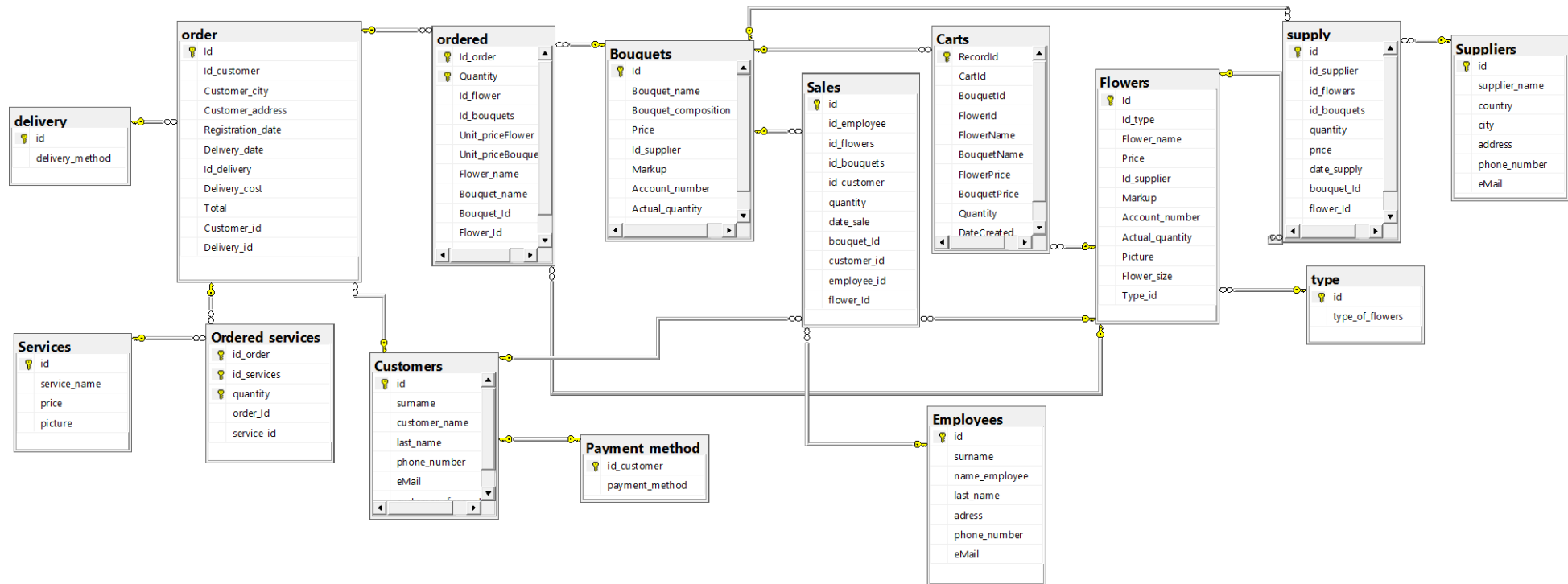


Рисунок 2.6 – Схема нормалізованої бази даних

Пояснення до схеми бази даних наведено у таблиці 13.

Таблиця 2.13 – Пояснення до схеми БД

Назва таблиці	Призначення
Bouquets	Містить інформацію про букети.
Carts	Кошик
Customers	Містить інформацію про клієнтів.
Delivery	Містить інформацію про доставку.
Employees	Містить інформацію про співробітників.
Flowers	Містить інформацію про квіти.
Order	Містить інформацію про замовлення.
Ordered	Містить інформацію про те, що замовлено.
Ordered_services	Містить інформацію про послугу, яку замовлено.
Payment_method	Містить інформацію про спосіб оплати.
Sales	Містить інформацію про продажі.
Services	Містить інформацію про послуги.
Suppliers	Містить інформацію про постачальників.
Supply	Містить інформацію про постачання.
Type	Містить інформацію про тип квітів.

Схема бази даних була сформована автоматчно за допомогою Microsoft SQL Server.

2.6 Діаграма пакетів (Package diagram)

Діаграма пакетів потрібна для організації елементів у групи за будь-якою ознакою із метою спрощення структури та організації роботи із моделлю системи.

Зображена на рисунку 2.7 діаграма пакетів відображає пакети класів та залежності між зазначеними пакетами.

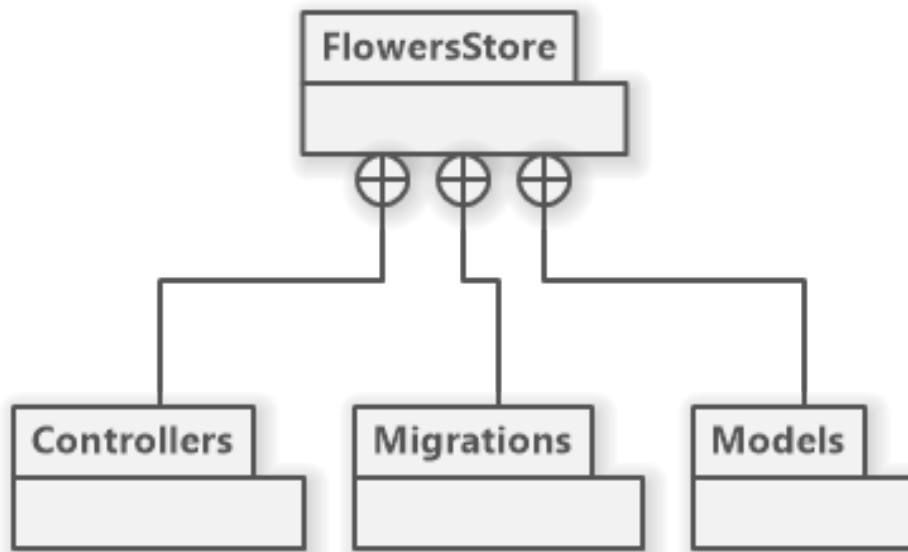


Рисунок 2.7 – Діаграма пакетів

Пакет (package) – це інструмент групування, який дозволяє взяти будь-яку конструкцію UML та об’єднати її елементи в одиниці високого рівня. Здебільшого пакети служать для об’єднання класів у групи.

Висновки до розділу 2

У результаті написання розділу 2 КРБ було наведено сценарії використання інтернет магазину з продажу букетів, а також створено діаграму використання для наочної демонстрації можливостей кожного з користувачів.

Далі було розроблено діаграму класів для визначення структури системи і виконано проектування бази даних на основі визначеної структури класів.

Також було розроблено діаграму станів, що дає змогу краще зрозуміти функції застосунку, а також діаграму пакетів, яка висвітлює архітектуру проекту. Для проектування було використано мову моделювання UML. Для створення діаграм знадобилося таке програмне забезпечення, як Lucidchart [15] та Software Ideas Modeler.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Обґрунтування вибору технологій для реалізації системи

Вибір програмних засобів для розробки

Розробка веб-застосунків вимагає ретельного вибору технологій і підходів. Важливо також зрозуміти, як вони були відібрані для проекту.

При виборі технології важливо враховувати наступне [18]:

- складність програмного забезпечення;
- наявність ресурсів;
- наявність готових компонентів;
- наявність технічної документації;
- характеристики якості пз;
- витрати на технологію;
- ліцензійну політику;
- вимоги безпеки.

У якості технології розробки програмного забезпечення було обрано ASP .NET MVC, а у якості бази даних було обрано SQL. Для цього було обрано відповідне програмне забезпечення – Visual Studio (для написання програмного коду) та MS SQL Server (для створення бази даних).

ASP .NET MVC – програмне середовище розробки гнучких, динамічних вебзастосунків та сервісів, дружня із принципами Test-Driven Development (розробка через тестування).

.NET – кросплатформне середовище із відкритим вихідним кодом для створення застосунків різних типів. Із .NET можна використовувати різні мови програмування, редактори та бібліотеки для написання мобільних, класичних та вебзастосунків, ігр та застосунків для Інтернету речей.

Архітектура MVC (Model-View-Controller) – розділяє застосунок на три основні групи компонентів; модель, вид та контролер.

На рисунку 3.1 зображена архітектура MVC.

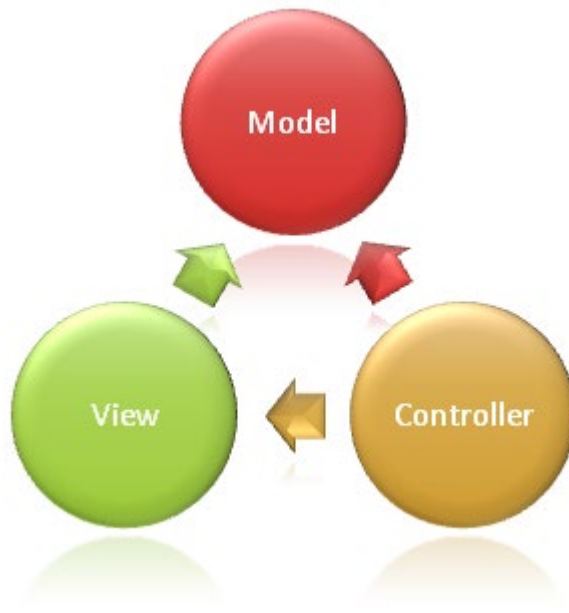


Рисунок 3.1 – Архітектура MVC.

Модель – це компонент програми, який працює з даними. Він зберігає інформацію та може відображати її в базі даних. Тобто він має можливість додавати записи до бази даних, видаляти їх, змінювати або просто отримувати їх. Завдання моделі – взяти дані та доставити їх користувачеві, яка їх запитала. Модель взаємодіє з контролером, як видно на рис. 3.1. У результаті контролер може приймати або передавати дані в модель. З іншого боку моделі зазвичай є база даних, де модель може зберігати та отримувати дані.

Представлення відповідає за відображення вмісту через інтерфейс користувача. Для цього використовується обробник представлення Razor для впровадження коду .NET і розмітку HTML. Представлення повинні мати мінімальну логіку, яка повинна бути пов'язана із відображенням вмісту. Якщо є необхідність відобразити більшу частину логіки для відображення даних із складної моделі, рекомендується використати компоненти представлення, ViewModel або шаблон представлення, який дозволяє спростити вид.

Контролер – компонент для управління взаємовідносинами з користувачем, роботою із моделлю та вибором представлення для відображення. Контролер обробляє введення даних, формує відповідь та взаємодіє з користувачем. Він є початковою відправною точкою та відповідає за вибір робочих типів моделей та

відображення представлення, тобто контролює яким саме чином відбувається відповідь на конкретний запит.

Для розробки вебзастосунків використовується дві мови програмування: ASP .NET та PHP. Порівняння цих мов наведено у таблиці 3.1.

Таблиця 3.1 – Порівняння ASP .NET та PHP [4]

	ASP .NET	PHP
Тип коду	Скомпільований код	Інтерпретований код
Масштабованість	Вертикальна	Горизонтальна
Функціональність	Більша кількість вбудованих особливостей	Менша кількість вбудованих особливостей
Вартість	Вища плата за хостинг	Нища плата за хостинг
Розмір	Середній та великий розмір	Маленький та середній розмір
Рішення	Безпека та функціональність	Інтерфейс користувача
Безпека	Висока	Більш низька
Швидкість	Більш швидкий для desktop застосунків	Більш повільний для desktop застосунків
Складність у вивченні	Більш складний	Більш легкий

Отже, і ASP .NET, і PHP мають свої переваги та недоліки. Використовувати потрібно технологію в залежності від поставленого завдання.

MS SQL Server – система управління реляційними базами даних. SQL Server характеризується такими особливостями, як:

- Продуктивність – працює дуже швидко;
- Надійний та безпечний – забезпечує шифрування даних;
- Простий – дуже легко працювати та адмініструвати.

Visual Studio – інтегроване середовище розробки програмного забезпечення, яка дає можливість редагувати, налагоджувати та збирати код, а також публікувати застосунок. Є підтримка багатьох об'єктно-орієнтованих мов програмування, що є перевагою.

3.2 Опис програмної реалізації

Підготовка до розробки програмного забезпечення

Перед тим, як почати розробку застосунку, потрібно встановити середовище розробки. Visual Studio має декілька версій: Community, Enterprise та Professional. Було обрано версію Community, оскільки вона є безкоштовною для студентів. Завантажується із офіційного сайту компанії Microsoft, яка є її розробником. При встановленні можна обрати окремі компоненти або встановити лише базові. Перевагою є те, що пізніше можна дозавантажити потрібні компоненти або навпаки видалити їх. На рисунку 3.2 позана сторінка встановлення додаткових компонентів.

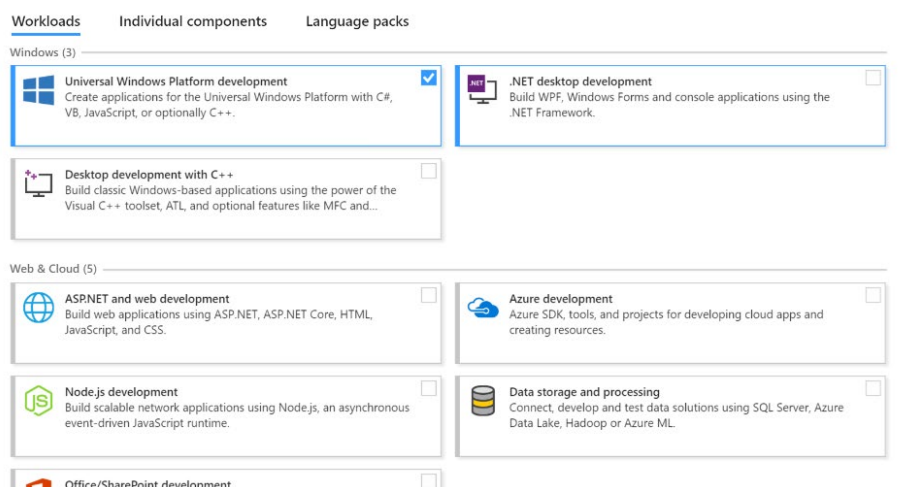


Рисунок 3.2 – встановлення додаткових компонентів

Також потрібно встановити MS SQL Server, який також встановлюється із сайту Microsoft. Існує дві версії – Developer та Express, було обрано версію Express. На рисунку 3.3 вигляд після встановлення MS SQL Server.

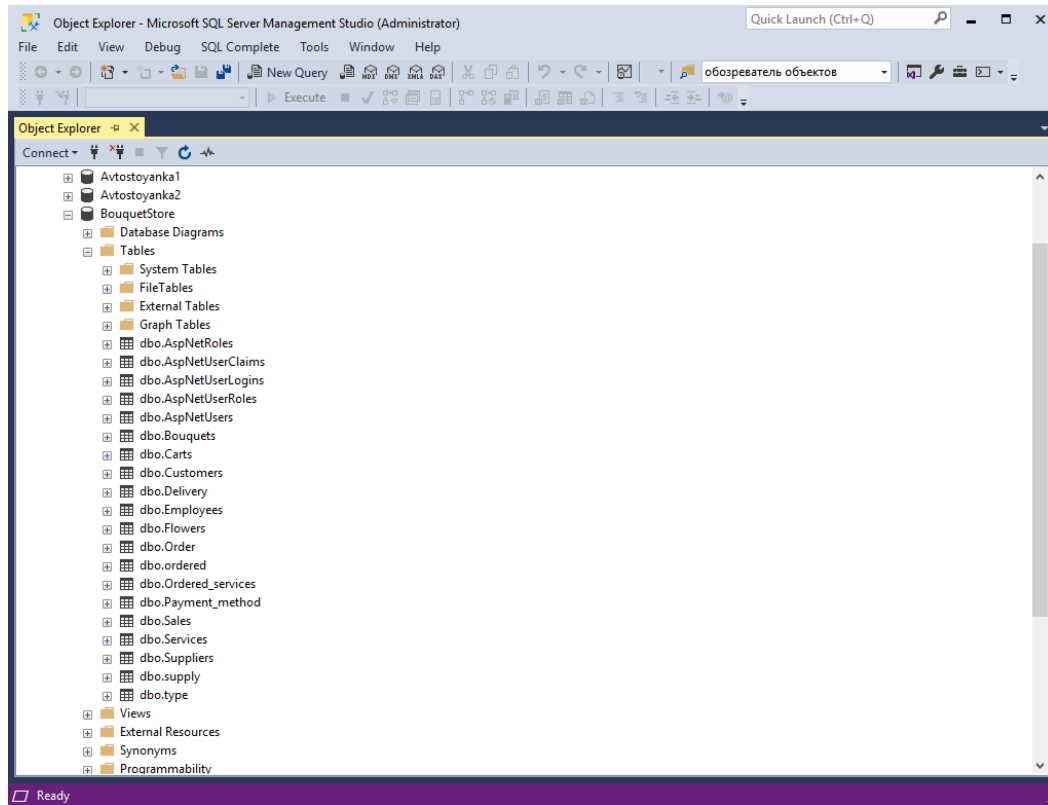


Рисунок 3.3 – MS SQL Server

Структура проекту

Після встановлення необхідного програмного забезпечення розглядається структура проекту.

Проект складається із наступних розділів:

Properties – призначений для зберігання профілів публікацій.

Посилання – призначений для зберігання всіх додатково встановлених бібліотек.

App_Data – призначений для зберігання бази даних після реалізації застосунку.

App_Start – призначений для основних налаштувань конфігурації, маршрутів та фільтрів.

Content – призначений для статичного розміщення картинок та CSS-файлів

Controllers – призначений для класів контролерів.

Fonts – призначений для зберігання шрифтів.

Migrations – призначений для міграцій бази даних.

Models – призначений для класів моделей.

Scripts – призначений для зберігання бібліотек JavaScript.

Views – призначений для представлень або часткових представлень. Зазвичай сгруповані за назвами контролерів з якими вони пов'язані.

Views/Shared – зберігаються компоненти, які не пов'язані із контролерами.

Та ще декількох файлів, таких як:

- Global.asax – глобальний клас, конфігурація маршрутів код, який повинен виконуватись при запуску та завершенні застосунку або під час виключень помилок;
- packages.config – пакети Microsoft.
- Startup.cs – файл конфігурації.
- Web.config – файл кофігурації.

Розробка основного функціоналу

У наступному коді наведено код контролеру HomeController, який відображає вміст основної сторінки.

Лістинг 3.1 – HomeController

```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }

    public ActionResult About()
    {
        ViewBag.Message = "Your application description page.";

        return View();
    }

    public ActionResult Contact()
    {
        ViewBag.Message = "Your contact page.";

        return View();
    }
}
```

```
public ActionResult Info()
{
    return View();
}
```

Код всіх інших контролерів винесено у додатки, оскільки він є великим.

Далі наведено код моделі квітів.

Лістинг 3.2 – Flower

```
public partial class Flower
{
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
"CA2214:DoNotCallOverridableMethodsInConstructors")]
    public Flower()
    {
        Ordereds = new HashSet<Ordered>();
        Sales = new HashSet<Sale>();
        Supplies = new HashSet<Supply>();
    }

    public int Id { get; set; }

    public short Id_type { get; set; }

    [Required]
    [StringLength(50)]
    public string Flower_name { get; set; }

    [Column(TypeName = "money")]
    public decimal Price { get; set; }

    public int Id_supplier { get; set; }

    public int Markup { get; set; }

    public int Account_number { get; set; }

    public int Actual_quantity { get; set; }

    public byte[] Picture { get; set; }

    public string PictureType { get; set; }

    public int Flower_size { get; set; }

    public virtual Type Type { get; set; }

    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
"CA2227:CollectionPropertiesShouldBeReadOnly")]
    public virtual ICollection<Ordered> Ordereds { get; set; }

    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
"CA2227:CollectionPropertiesShouldBeReadOnly")]
    public virtual ICollection<Sale> Sales { get; set; }

    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
"CA2227:CollectionPropertiesShouldBeReadOnly")]
    public virtual ICollection<Supply> Supplies { get; set; }
```

}

Також наведено код моделі кошику покупок.

Лістинг 3.3 – Cart

```
public partial class Cart
{
    [Key]
    public int RecordId { get; set; }
    public int CartId { get; set; }
    public int BouquetId { get; set; }
    public int FlowerId { get; set; }
    public string FlowerName { get; set; }
    public string BouquetName { get; set; }
    public decimal FlowerPrice { get; set; }
    public decimal BouquetPrice { get; set; }
    public float Bill { get; set; }
    public int Quantity { get; set; }
    public DateTime DateCreated { get; set; }
    public byte[] Picture { get; set; }
    public virtual Bouquet Bouquet { get; set; }
    public virtual Flower Flower { get; set; }
}
```

Код всіх інших моделей є аналогічним, окрім тих, що призначені для реєстрації користувачів (вони будуть винесені у додатки).

Розробка додаткового функціоналу

У застосунку є підтвердження реєстрації за допомогою пошти. Нижче наведено код.

Лістинг 3.4 – Підтвердження реєстрації за допомогою пошти

```
public class EmailService : IIdentityMessageService
{
    public Task SendAsync(IdentityMessage message)
    {
        var apiKey = Environment.GetEnvironmentVariable("Yuliya");
        var client = new SendGridClient(apiKey);
        var from = new EmailAddress("elinaazaliya@gmail.com", "Yuliya");
        var subject = message.Subject;
        var to1 = new EmailAddress(message.Destination);
        var to = message.Destination;
        var plainTextContent = message.Body;
        var htmlContent = message.Body;
        var msg = MailHelper.CreateSingleEmail(from, to1, subject, plainTextContent,
htmlContent);
        var response = client.SendEmailAsync(msg);
        return response;
    }
}
```

Код підтвердження реєстрації за номером телефону.

Лістинг 3.5 – Підтвердження реєстрації за допомогою телефону

```
public class SmsService : IIdentityMessageService
{
    public Task SendAsync(IdentityMessage message)
    {
        // Twilio Begin
        var accountSid = ConfigurationManager.AppSettings["SMSAccountIdentification"];
        var authToken = ConfigurationManager.AppSettings["SMSAccountPassword"];
        var fromNumber = ConfigurationManager.AppSettings["SMSAccountFrom"];

        TwilioClient.Init(accountSid, authToken);

        MessageResource result = MessageResource.Create(
            new PhoneNumber(message.Destination),
            from: new PhoneNumber(fromNumber),
            body: message.Body
        );

        valid //Status is one of Queued, Sending, Sent, Failed or null if the number is not

        Trace.TraceInformation(result.Status.ToString());
        //Twilio doesn't currently have an async API, so return success.
        // Подключите здесь службу SMS, чтобы отправить текстовое сообщение.
        return Task.FromResult(0);
    }
}
```

У застосунку також представлено адміністрування користувачів. Наприклад, користувач, який не є адміністратором, не має доступу до редагування, додавання або видалення квітів із магазину. Його дії обмежуються за допомогою:

```
@if (Request.IsAuthenticated && User.IsInRole("Admin"))
```

3.3 Керівництво користувача

Склад проекту

Даний вебсайт складається із двох частин: клієнтська частина та адміністративна панель.

Призначенням клієнтської частини є перегляд контенту сайту та замовлення квітів. Ця частина доступна всім користувачам.

Адміністративна частина доступна лише адміністратору, який може змінювати інформацію, тобто редагувати, додавати або змінювати.

Клієнтська частина

При відкритті сайту в браузері відкривається головна сторінка (рисунок 3.4), на шапці якої розташовані посилання на сторінку авторизації/реєстрації, вкладку «Про нас», «Контактні дані», кошик, а також логотип сайту. Оскільки сторінок сайту небагато, то посилання для користувачів посилання на інші сторінки сайту містяться також на шапці.

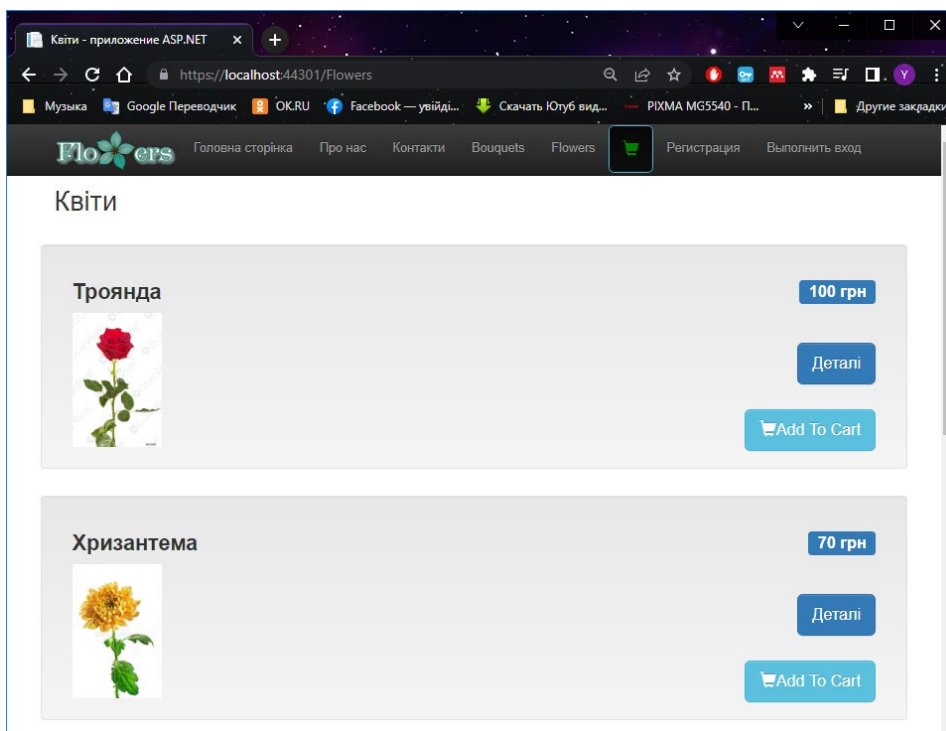


Рисунок 3.4 – Головна сторінка

Для адміністраторів створена навігаційна панель.

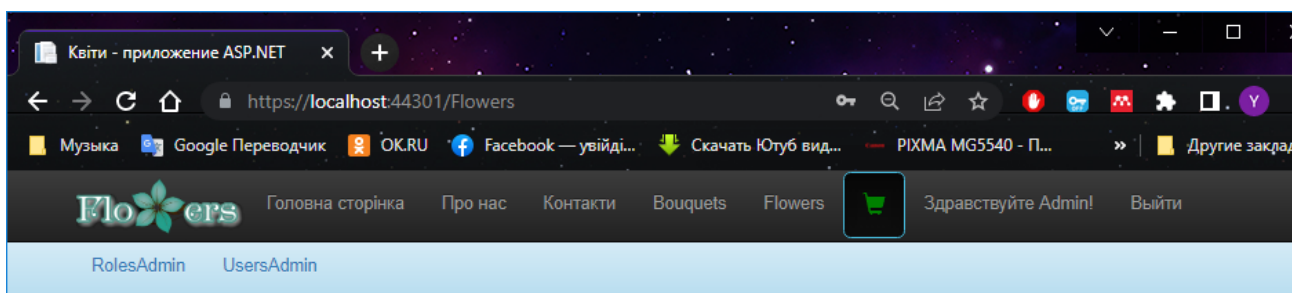


Рисунок 3.5 – Навігаційна панель

При натисканні на кнопку із назвою категорії відкривається сторінка із певним товаром з обраної категорії.

На сторінках із товаром міститься назва, фото та ціна товару.

Кожен товар можна додати до кошику та повернутись до продовження покупок або перегляду контенту.

При натисненні кнопки «Додати у кошик» відкривається сторінка із підтвердженням, а також можна вказати кількість товару для додавання у кошик.

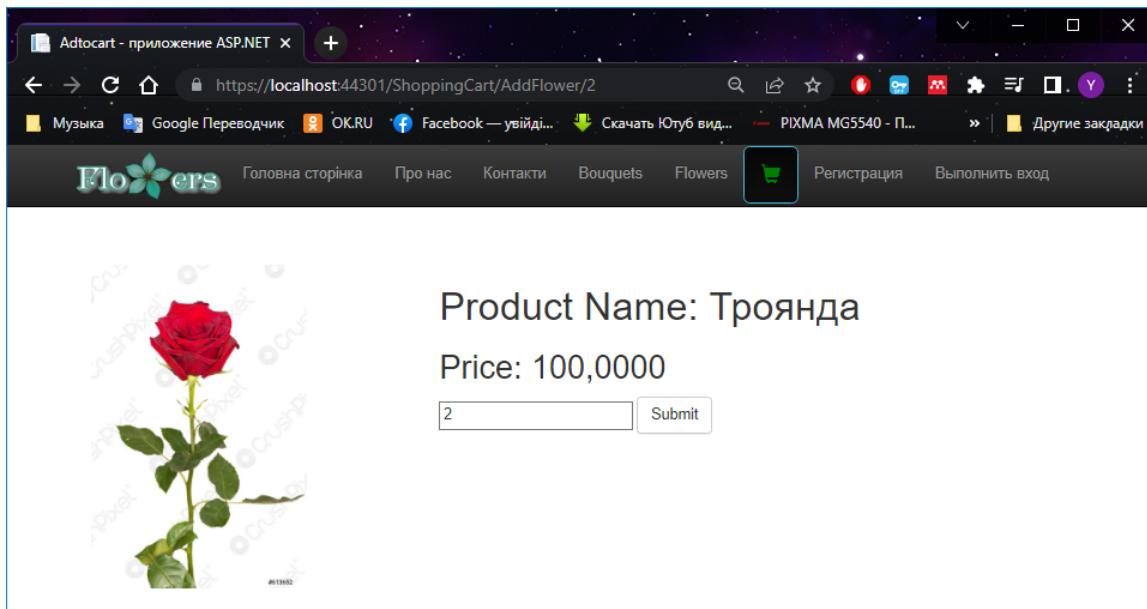


Рисунок 3.6 – Сторінка підтвердження

Щоб переглянути вміст кошику потрібно натиснути на кошик, що знаходиться на шапці сайту. Під списком товарів знаходиться загальна сума замовлення.

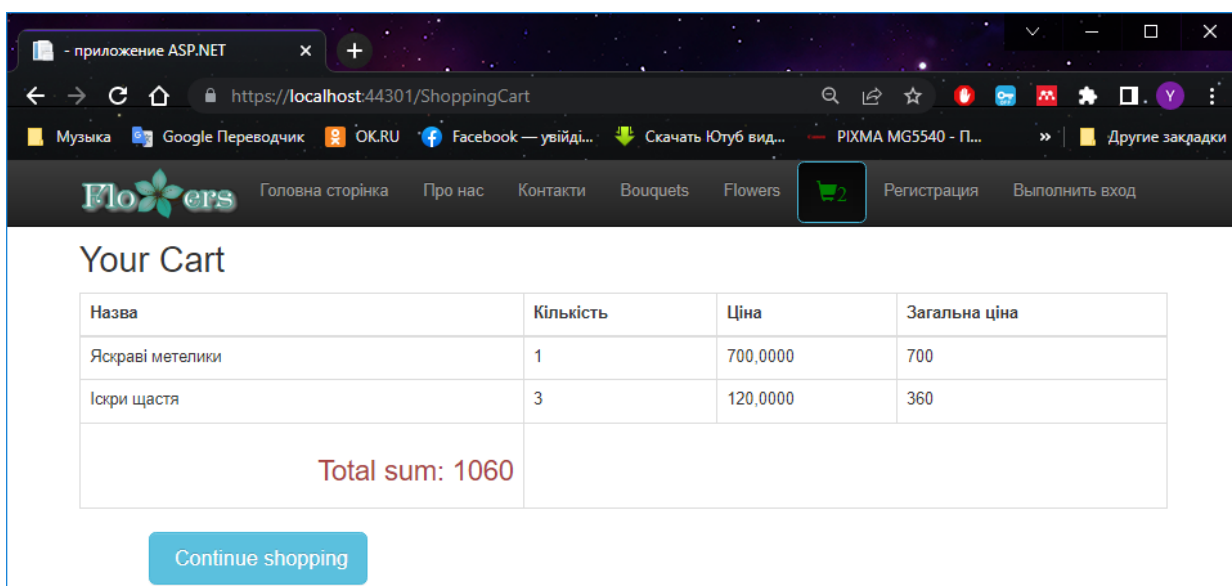


Рисунок 3.7 – Вміст кошику

Також при додаванні товару біля кошику з'являється цифра із кількістю доданих товарів.

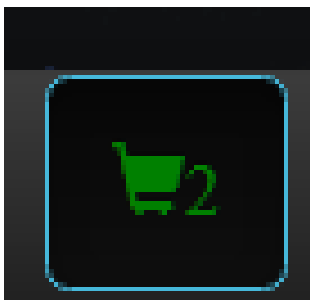


Рисунок 3.8 – Кількість доданих товарів

Сторінка про нас містить детальний опис інтернет магазину.

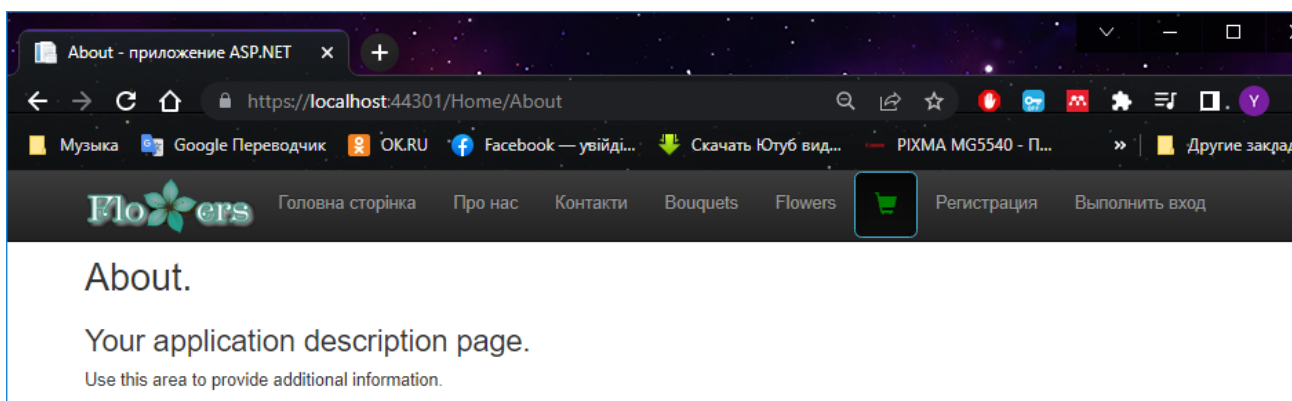


Рисунок 3.9 – Сторінка «Про нас»

Для входу на сайт потрібно ввести email та пароль.

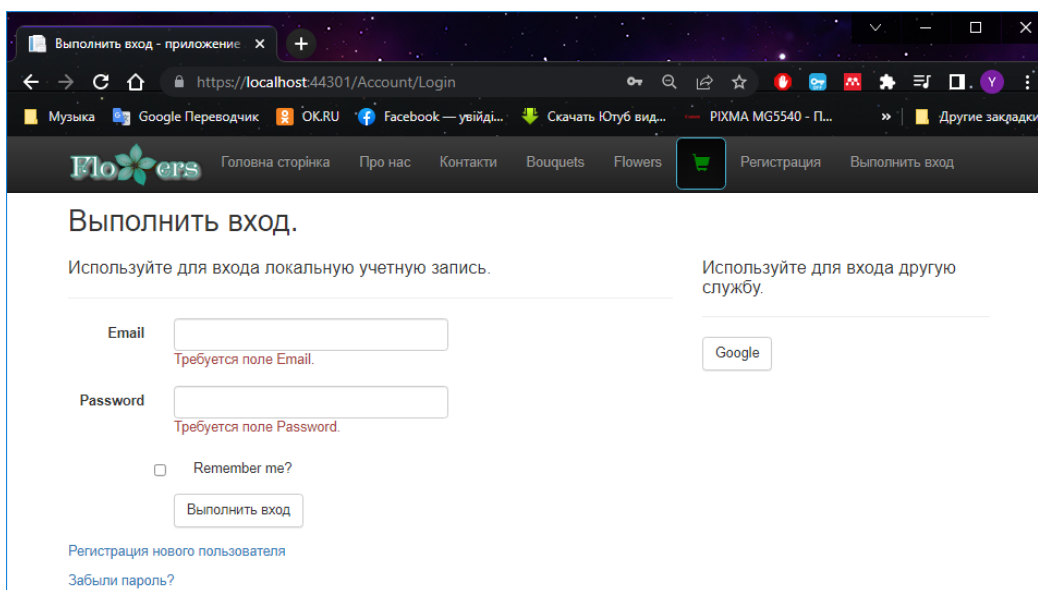


Рисунок 3.10 – Сторінка входу

Для реєстрації потрібно ввести ім'я, email, пароль та повторно пароль, які повинні співпасти.

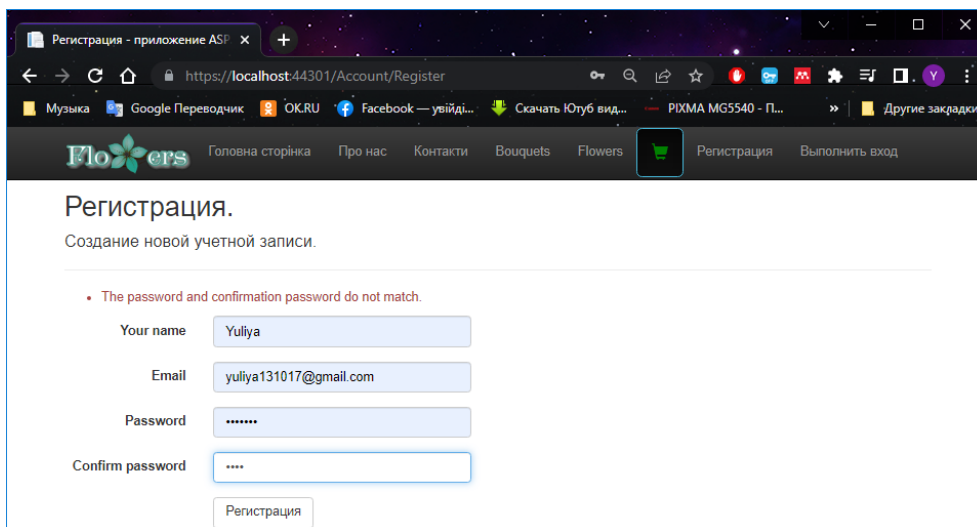


Рисунок 3.11 – Сторінка реєстрації

Далі потрібно підтвердити реєстрацію, лист приходять на пошту.

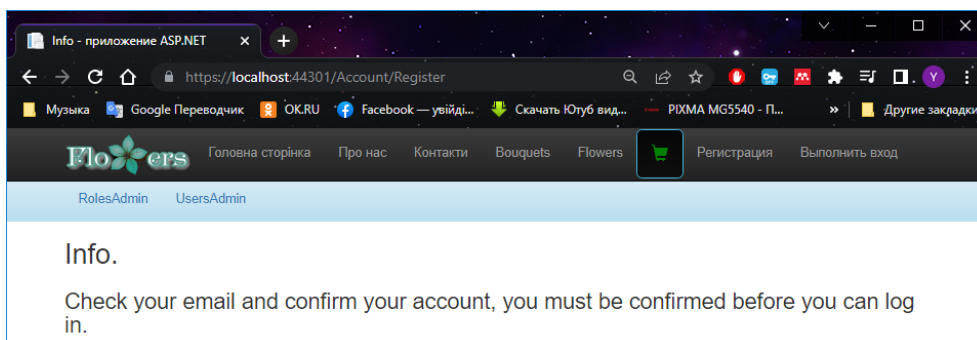


Рисунок 3.12 – Відправка листа

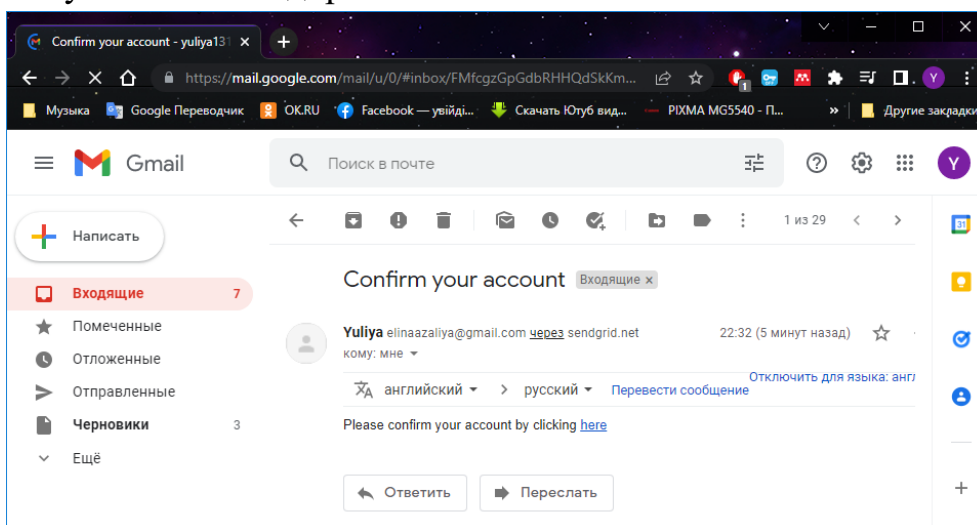


Рисунок 3.13 – Лист на пошті

Також користувач може ввімкнути двухфакторну авторизацію та підтвердити за допомогою пошти або номеру телефону.

Ще можна автоізуватись за допомогою Google.

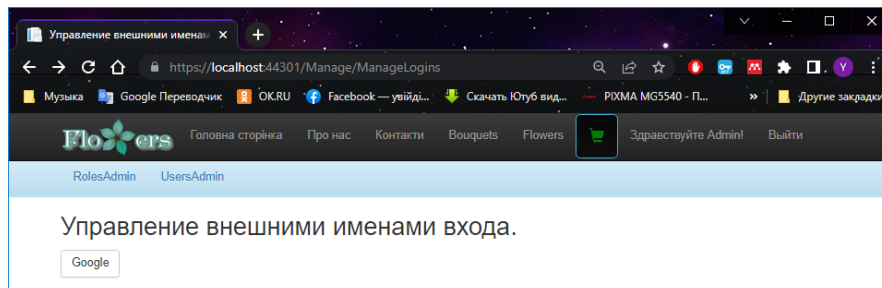


Рисунок 3.12 – Авторизація за допомогою Google

Адміністративна панель

Щоб зайти до адміністративної сторінки повинні бути права адміністратора та авторизуватись як адміністратор.

Після реєстрації як адміністратор користувач потрапляє на сторінку адміністративної панелі.

Сторінка редагування товару на рисунку 3.13

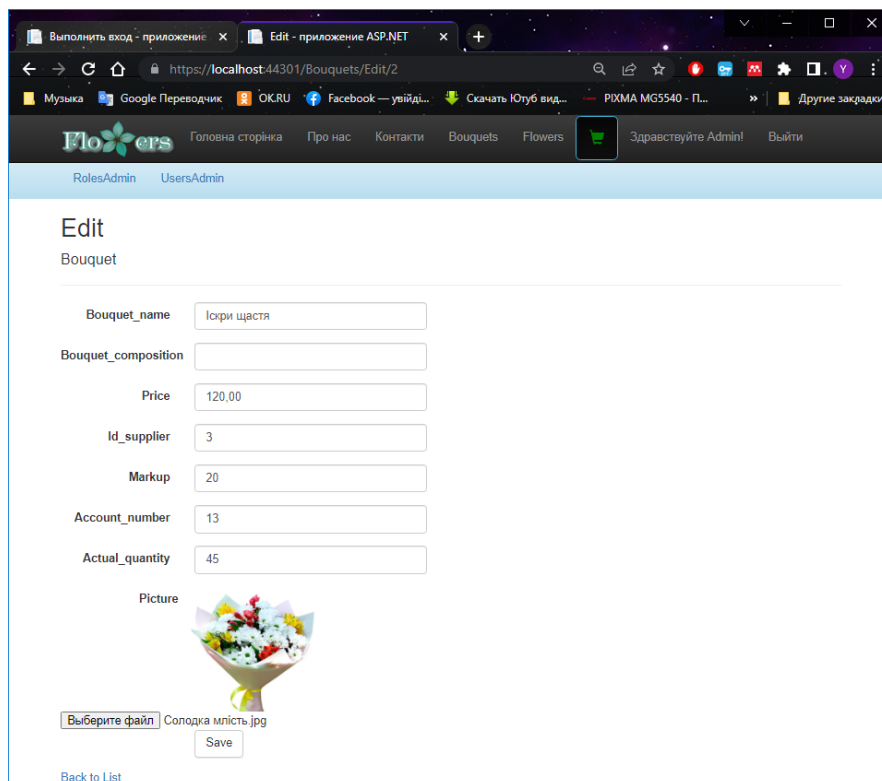


Рисунок 3.13 – Сторінка редагування товару

Сторінка додавання товару на рисунку 3.14

Кафедра інженерії програмного забезпечення
Веб застосунок магазину продажу квітів


The screenshot shows a web browser window with the URL `https://localhost:44301/Bouquets/Create/2`. The page title is "Create Bouquet". The form contains the following fields:

- Bouquet_name:** Подих весни
- Bouquet_composition:** Тюльпани
- Price:** 565
- Id_supplier:** 5
- Markup:** 14
- Account_number:** 5
- Actual_quantity:** 10
- Picture:** Выберите файл Яскраві метелики.jpg

A "Create" button is located at the bottom of the form.

Рисунок 3.14 – Сторінка додавання товару
Сторінка підтвердження видалення на рисунку 3.15

The screenshot shows a web browser window with the URL `https://localhost:44301/Bouquets/Delete/2`. The page title is "Delete". The confirmation message is "Are you sure you want to delete this?". The bouquet details are as follows:

Bouquet_name	Іскри щастя
Bouquet_composition	
Price	120,00
Id_supplier	3
Markup	20
Account_number	13
Actual_quantity	45
Picture	

At the bottom, there are two buttons: "Delete" and "Back to List".

Рисунок 3.15 – Сторінка підтвердження видалення

Адміністратор може надати користувачеві роль адміністратора або видалити користувача із цієї ролі.

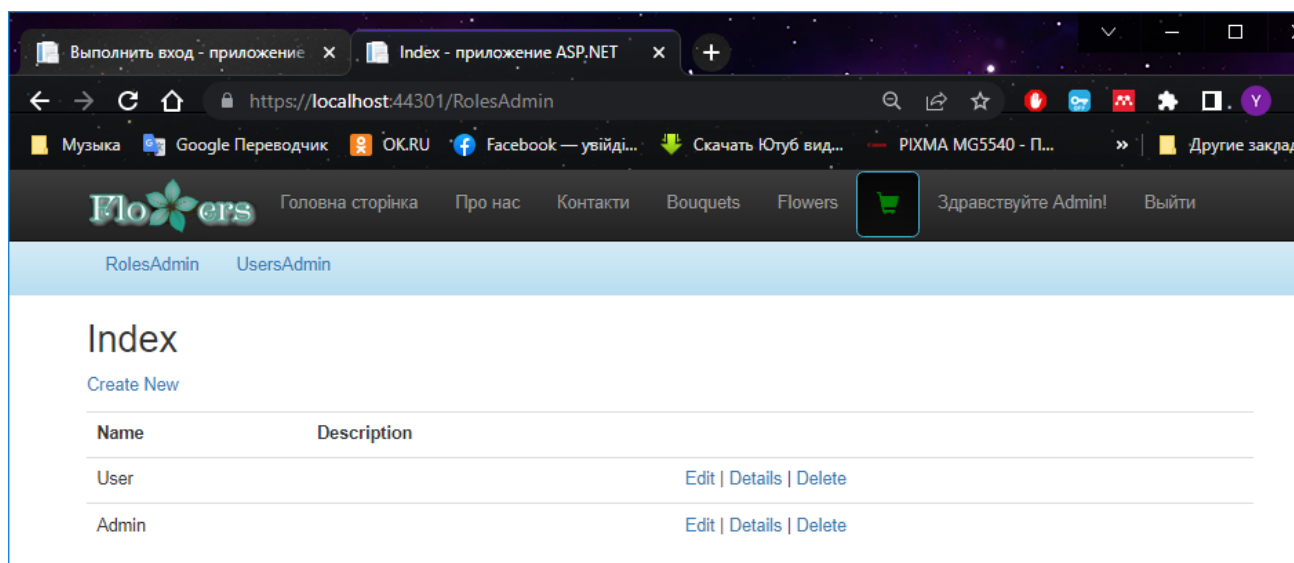


Рисунок 3.16 – Адміністрування користувачів

Також адміністратор може редагувати користувача, тобто змінювати його роль.

Висновки до розділу 3

У результаті написання розділу 3 КРБ було обґрунтовано вибір технологій для реалізації системи. З'ясовано, які фактори впливають на вибір технологій.

Кожна технологія має свої переваги та недоліки, які враховано при виборі. Представлено структуру проекту, який складається із двох частин: клієнтської та адміністративної панелі.

Проект має основний та додатковий функціонал. Додатковий функціонал складається із двухфакторної авторизації та деяких обмежень. Також представлено керівництво користувача.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи бакалавра було розроблено вебзастосунок Інтернет-магазину продажу квітів. Даний проєкт надає можливість продавати квіти та букети квітів в мережі інтернет.

Для досягнення поставленої мети було виконано **наступні завдання:**

- 1) Проаналізовано аналогічні системи, виявлено їх переваги та недоліки;
- 2) Створено технічне завдання (ТЗ);
- 3) Проведено аналіз вимог до системи;
- 4) Проведено аналіз системи;
- 5) Здійснено проектування системи;
- 6) Розроблено інтернет магазин.

Було наведено сценарії використання вебзастосунку, а також створено діаграму класів, діаграми станів та пакетів для наочної роботи застосунку та його функцій. Також розроблено та створено схему бази даних.

Також представлено інтерфейс системи.

В процесі роботи усунуто головний недолік аналогів – оптимізовано інтерфейс за рахунок зменшення кількості модальних вікон з рекламою та акціями.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Account Confirmation & Password Recovery - ASP.NET Identity (C#) - ASP.NET 4.x | Microsoft Docs. URL: <https://docs.microsoft.com/en-us/aspnet/identity/overview/features-api/account-confirmation-and-password-recovery-with-aspnet-identity> (accessed 19/06/2022).
2. ASP.NET MVC 5 app with SMS and email Two-Factor Authentication | Microsoft Docs. URL: <https://docs.microsoft.com/en-us/aspnet/mvc/overview/security/aspnet-mvc-5-app-with-sms-and-email-two-factor-authentication> (accessed 27/04/2022).
3. ASP.NET MVC Diagram UML Class Diagram Example - Syncfusion Demos. URL: <https://ej2.syncfusion.com/aspnetmvc/Diagram/UmlClassDiagram#/fluent> (accessed 19/06/2022).
4. ASP.NET vs PHP: Which is Better for Web Development? - Existek Blog. URL: <https://existek.com/blog/aspnet-vs-php-which-is-better-for-web-development/> (accessed 19/06/2022).
5. AspNet-Identity-2-Extending-Users-And-Roles/AccountController.cs at master · TypecastException/AspNet-Identity-2-Extending-Users-And-Roles. URL: <https://github.com/TypecastException/AspNet-Identity-2-Extending-Users-And-Roles/blob/master/AspNetIdentity2ExtendingUsersAndRoles/Controllers/AccountController.cs> (accessed 27/04/2022).
6. BuiltWith технологический запрос. URL: <https://builtwith.com/ru/> (accessed 19/05/2022).
7. Code First в существующую базу данных — EF6 | Microsoft Docs. URL: <https://docs.microsoft.com/ru-ru/ef/ef6/modeling/code-first/workflows/existing-database> (accessed 19/06/2022).
8. Create a secure ASP.NET MVC 5 web app with log in, email confirmation and password reset (C#) | Microsoft Docs. URL: <https://docs.microsoft.com/en-us/aspnet/mvc/overview/security/create-an-aspnet-mvc-5-web-app-with-email->

- confirmation-and-password-reset (accessed 19/06/2022).
9. Difference between PHP and ASP.NET - GeeksforGeeks. URL: <https://www.geeksforgeeks.org/difference-between-php-and-asp-net/> (accessed 19/06/2022).
 10. Microsoft Corporation Shopping Cart | Microsoft Docs. 2014. URL: <https://docs.microsoft.com/en-us/aspnet/web-forms/overview/getting-started/getting-started-with-aspnet-45-web-forms/shopping-cart> (accessed 27/04/2022).
 11. Part 9: Registration and Checkout | Microsoft Docs. URL: <https://docs.microsoft.com/en-us/aspnet/mvc/overview/older-versions/mvc-music-store/mvc-music-store-part-9> (accessed 27/04/2022).
 12. SendGrid. URL: <https://app.sendgrid.com/guide/integrate/langs/csharp> (accessed 19/06/2022).
 13. Sql server, .net and c# video tutorial: ASP.NET Web API google authentication. URL: <https://csharp-video-tutorials.blogspot.com/2016/12/aspnet-web-api-google-authentication.html?m=1> (accessed 27/04/2022).
 14. State diagram with example - YouTube. URL: <https://www.youtube.com/watch?v=L9UCsQxuWmw> (accessed 19/06/2022).
 15. UML діаграмми онлайн | Lucidchart. URL: <https://www.lucidchart.com/pages/ru> (accessed 19/05/2022).
 16. Диаграммы UML для моделирования процессов и архитектуры проекта. URL: <https://evergreens.com.ua/ru/articles/uml-diagrams.html> (accessed 19/05/2022).
 17. Как и зачем писать Use Cases | DOU. URL: <https://dou.ua/lenta/articles/use-cases/> (accessed 14/06/2022).
 18. Лекція 5 - Основи програмної інженерії. URL: <https://sites.google.com/site/osnoviprogramnoieinzeneriie/lekcii/lekcia-6> (accessed 17/06/2022).

ДОДАТОК А

BOUQUETS CONTROLLER

```

public class BouquetsController : Controller
{
    private FlowersStoreDB db = new FlowersStoreDB();
    public Bouquet Bouquet
    {
        get => default;
        set { }
    }
    // GET: Bouquets
    public ActionResult Index()
    {
        return View(db.Bouquets.ToList());
    }
    // GET: Bouquets/Details/5
    public ActionResult Details(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Bouquet bouquet = db.Bouquets.Find(id);
        if (bouquet == null)
        {
            return HttpNotFound();
        }
        return View(bouquet);
    }
    // GET: Bouquets/Create
    public ActionResult Create()
    {
        return View();
    }
    // POST: Bouquets/Create
    // Чтобы защититься от атак чрезмерной передачи данных, включите определенные
    // свойства, для которых следует установить привязку.
    // Дополнительные сведения см. в статье
    https://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create([Bind(Include =
    "id,bouquet_name,bouquet_composition,price,id_supplier,markup,account_number,actual_quantity,p
    icture")] Bouquet bouquet)
    {
        if (ModelState.IsValid)
        {
            db.Bouquets.Add(bouquet);
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        return View(bouquet);
    }
    // GET: Bouquets/Edit/5
    public ActionResult Edit(int? id)
    {
        if (id == null)
        {

```

```

        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Bouquet bouquet = db.Bouquets.Find(id);
    if (bouquet == null)
    {
        return HttpNotFound();
    }
    return View(bouquet);
}
// POST: Bouquets/Edit/5
// Чтобы защититься от атак чрезмерной передачи данных, включите определенные
свойства, для которых следует установить привязку. Дополнительные сведения см. в статье
// https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"id,bouquet_name,bouquet_composition,price,id_supplier,markup,account_number,actual_quantity,p
icture")]) Bouquet bouquet)
{
    if (ModelState.IsValid)
    {
        db.Entry(bouquet).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(bouquet);
}
// GET: Bouquets/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Bouquet bouquet = db.Bouquets.Find(id);
    if (bouquet == null)
    {
        return HttpNotFound();
    }
    return View(bouquet);
}
// POST: Bouquets/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Bouquet bouquet = db.Bouquets.Find(id);
    db.Bouquets.Remove(bouquet);
    db.SaveChanges();
    return RedirectToAction("Index");
}
protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
}

```


ДОДАТОК Б

SHOPPING CART CONTROLLER

```

public class ShoppingCartController : Controller
{
    FlowersStoreDB db = new FlowersStoreDB();

    List<Cart> li = new List<Cart>();

    public ActionResult IndexFlower()
    {
        if (TempData["cart"] != null)
        {
            float x = 0;
            List<Cart> li2 = TempData["cart"] as List<Cart>;
            foreach (var item in li2)
            {
                x += item.Bill;
            }

            TempData["total"] = x;
        }
        TempData.Keep();

        return View(db.Flowers.OrderByDescending(x => x.Id).ToList());
    }

    public ActionResult AddFlower(int? id)
    {
        Flower flower = db.Flowers.Where(x => x.Id == id).SingleOrDefault();
        return View(flower);
    }

    [HttpPost]
    public ActionResult AddFlower(int quantity, int id)
    {
        Flower flower = db.Flowers.Where(x => x.Id == id).SingleOrDefault();

        Cart cart = new Cart();
        cart.FlowerName = flower.Flower_name;
        cart.FlowerPrice = flower.Price;
        cart.Quantity = quantity;
        cart.Bill = (float)(cart.FlowerPrice * cart.Quantity);

        if (TempData["cart"] == null)
        {
            li.Add(cart);
            TempData["cart"] = li;
            Session["count"] = 1;
        }
        else
        {
            List<Cart> li2 = TempData["cart"] as List<Cart>;
            li2.Add(cart);
            TempData["cart"] = li2;
            Session["count"] = Convert.ToInt32(Session["count"]) + 1;
        }

        TempData.Keep();
    }
}

```

Кафедра інженерії програмного забезпечення
Веб застосунок магазину продажу квітів

```

        return RedirectToAction("Index");
    }

    public ActionResult Index()
    {
        if (TempData["cart"] != null)
        {
            float x = 0;
            List<Cart> li2 = TempData["cart"] as List<Cart>;
            foreach (var item in li2)
            {
                x += item.Bill;
            }

            TempData["total"] = x;
        }
        TempData.Keep();

        List<Flower> flowers = db.Flowers.OrderByDescending(c => c.Id).ToList();
        List<Bouquet> bouquets = db.Bouquets.OrderByDescending(c => c.Id).ToList();

        return View(Tuple.Create(flowers, bouquets));
    }

    public ActionResult AddBouquet(int? id)
    {
        Bouquet bouquet = db.Bouquets.Where(x => x.Id == id).SingleOrDefault();
        return View(bouquet);
    }

    [HttpPost]
    public ActionResult AddBouquet(int quantity, int id)
    {
        Bouquet bouquet = db.Bouquets.Where(x => x.Id == id).SingleOrDefault();

        Cart cart = new Cart();

        cart.BouquetName = bouquet.Bouquet_name;
        cart.BouquetPrice = bouquet.Price;
        cart.Quantity = quantity;
        cart.Bill = (float)(cart.BouquetPrice * cart.Quantity);

        if (TempData["cart"] == null)
        {
            li.Add(cart);
            TempData["cart"] = li;
            Session["count"] = 1;
        }
        else
        {
            List<Cart> li2 = TempData["cart"] as List<Cart>;
            li2.Add(cart);
            TempData["cart"] = li2;
            Session["count"] = Convert.ToInt32(Session["count"]) + 1;
        }
        TempData.Keep();

        return RedirectToAction("Index");
    }
}

```

ДОДАТОК В

USERS ADMIN CONTROLLER

```
// [Authorize(Roles = "Admin")]
public class UsersAdminController : Controller
{
    public UsersAdminController()
    {
    }

    public UsersAdminController(ApplicationUserManager userManager, ApplicationRoleManager
roleManager)
    {
        UserManager = userManager;
        RoleManager = roleManager;
    }

    private ApplicationUserManager _userManager;
    public ApplicationUserManager UserManager
    {
        get
        {
            return _userManager ??
HttpContext.GetOwinContext().GetUserManager<ApplicationUserManager>();
        }
        private set
        {
            _userManager = value;
        }
    }

    private ApplicationRoleManager _roleManager;
    public ApplicationRoleManager RoleManager
    {
        get
        {
            return _roleManager ??
HttpContext.GetOwinContext().Get<ApplicationRoleManager>();
        }
        private set
        {
            _roleManager = value;
        }
    }

    //
    // GET: /Users/
    public async Task<ActionResult> Index()
    {
        return View(await UserManager.Users.ToListAsync());
    }

    //
    // GET: /Users/Details/5
    public async Task<ActionResult> Details(string id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
    }
}
```

Кафедра інженерії програмного забезпечення
Веб застосунок магазину продажу квітів

```

    }
    var user = await UserManager.FindByIdAsync(id);

    ViewBag.RoleNames = await UserManager.GetRolesAsync(user.Id);

    return View(user);
}

//
// GET: /Users/Create
public async Task<ActionResult> Create()
{
    //Get the list of Roles
    ViewBag.RoleId = new SelectList(await RoleManager.Roles.ToListAsync(), "Name",
"Name");
    return View();
}

//
// POST: /Users/Create
[HttpPost]
public async Task<ActionResult> Create(RegisterViewModel userModel, params
string[] selectedRoles)
{
    if (ModelState.IsValid)
    {
        var user = new ApplicationUser
        {
            UserName = userModel.Email,
            Email = userModel.Email,
        };

        // Then create:
        var adminresult = await UserManager.CreateAsync(user, userModel.Password);

        //Add User to the selected Roles
        if (adminresult.Succeeded)
        {
            if (selectedRoles != null)
            {
                var result = await UserManager.AddToRolesAsync(user.Id,
selectedRoles);
                if (!result.Succeeded)
                {
                    ModelState.AddModelError("", result.Errors.First());
                    ViewBag.RoleId = new SelectList(await
RoleManager.Roles.ToListAsync(), "Name", "Name");
                    return View();
                }
            }
        }
        else
        {
            ModelState.AddModelError("", adminresult.Errors.First());
            ViewBag.RoleId = new SelectList(RoleManager.Roles, "Name", "Name");
            return View();
        }
    }
    return RedirectToAction("Index");
}
ViewBag.RoleId = new SelectList(RoleManager.Roles, "Name", "Name");

```

Кафедра інженерії програмного забезпечення
Веб застосунок магазину продажу квітів

```

    return View();
}

//
// GET: /Users/Edit/1
public async Task<ActionResult> Edit(string id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    var user = await UserManager.FindByIdAsync(id);
    if (user == null)
    {
        return HttpNotFound();
    }

    var userRoles = await UserManager.GetRolesAsync(user.Id);

    return View(new EditUserViewModel()
    {
        Id = user.Id,
        Email = user.Email,
        RolesList = RoleManager.Roles.ToList().Select(x => new SelectListItem()
        {
            Selected = userRoles.Contains(x.Name),
            Text = x.Name,
            Value = x.Name
        })
    });
}

//
// POST: /Users/Edit/5
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Edit([Bind(Include =
"Email,Id,Address,City,State,PostalCode")] EditUserViewModel editUser, params string[]
selectedRole)
{
    if (ModelState.IsValid)
    {
        var user = await UserManager.FindByIdAsync(editUser.Id);
        if (user == null)
        {
            return HttpNotFound();
        }

        user.UserName = editUser.Email;
        user.Email = editUser.Email;

        var userRoles = await UserManager.GetRolesAsync(user.Id);

        selectedRole = selectedRole ?? new string[] { };

        var result = await UserManager.AddToRolesAsync(user.Id,
selectedRole.Except(userRoles).ToArray<string>());

        if (!result.Succeeded)
        {
            ModelState.AddModelError("", result.Errors.First());
        }
    }
}

```

```

        return View();
    }
    result = await UserManager.RemoveFromRolesAsync(user.Id,
userRoles.Except(selectedRole).ToArray<string>());

    if (!result.Succeeded)
    {
        ModelState.AddModelError("", result.Errors.First());
        return View();
    }
    return RedirectToAction("Index");
}
ModelState.AddModelError("", "Something failed.");
return View();
}

// GET: /Users/Delete/5
public async Task<ActionResult> Delete(string id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    var user = await UserManager.FindByIdAsync(id);
    if (user == null)
    {
        return HttpNotFound();
    }
    return View(user);
}

// POST: /Users/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<ActionResult> DeleteConfirmed(string id)
{
    if (ModelState.IsValid)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }

        var user = await UserManager.FindByIdAsync(id);
        if (user == null)
        {
            return HttpNotFound();
        }
        var result = await UserManager.DeleteAsync(user);
        if (!result.Succeeded)
        {
            ModelState.AddModelError("", result.Errors.First());
            return View();
        }
        return RedirectToAction("Index");
    }
    return View();
}
}
}

```

ДОДАТОК Г

ROLE VIEW MODEL

```
public class RoleViewModel
{
    public string Id { get; set; }

    [Required(AllowEmptyStrings = false)]
    [Display(Name = "RoleName")]

    public string Name { get; set; }
    public string Description { get; set; }
}

public class EditUserViewModel
{
    public string Id { get; set; }

    [Required]
    [Display(Name = "Your name")]
    public string UserName { get; set; }

    [Required(AllowEmptyStrings = false)]
    [Display(Name = "Email")]
    [EmailAddress]
    public string Email { get; set; }

    public IEnumerable<SelectListItem> RolesList { get; set; }
}
```

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

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА
ВЕБЗАСТОСУНОК МАГАЗИНУ ПРОДАЖУ КВІТІВ

СПЕЦІАЛЬНА ЧАСТИНА З ОХОРОНИ ПРАЦІ
ОХОРОНА ПРАЦІ ПРИ КОРИСТУВАННІ ЕКРАННИМИ ПРИСТРОЯМИ

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1– 408.1810802

Студент

_____ Ю.А. Андреева
підпис

«__» _____ 2022 р.

Консультант канд. техн. наук, доцент

_____ А. О. Алексеева
підпис

«__» _____ 2022 р.

Миколаїв – 2022

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
1 ГІГІЄНИЧНІ ВИМОГИ ДО ОРГАНІЗАЦІЇ РОБОТИ З ВІЗУАЛЬНИМИ ДИСПЛЕЙНИМИ ТЕРМІНАЛАМИ.....	5
1.1 Вимоги до виробничих приміщень для експлуатації ВДТ ЕОМ та ПЕОМ....	5
2 ГІГІЄНИЧНІ ВИМОГИ ДО ПАРАМЕТРІВ ВИРОБНИЧОГО СЕРЕДОВИЩА ПРИМІЩЕНЬ З ВДТ ЕОМ ТА ПЕОМ	7
2.1 Мікроклімат	7
2.1 Освітлення.....	7
2.3 Шум і вібрація	9
2.4 Неіонізуючі електромагнітні випромінювання.....	10
2.5 Іонізуючі електромагнітні випромінювання	10
3 ГІГІЄНИЧНІ ВИМОГИ ДО ОРГАНІЗАЦІЇ І ОБЛАДНАННЯ РОБОЧИХ МІСЦЬ З ВДТ ЕОМ І ПЕОМ	11
4 ВИМОГИ ДО РЕЖИМІВ ПРАЦІ І ВІДПОЧИНКУ ПРИ РОБОТІ З ВДТ ЕОМ І ПЕОМ	14
ВИСНОВКИ.....	17
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	18

ПЕРЕЛІК СКОРОЧЕНЬ

ВДТ – Візуальний дисплейний термінал

ЕОМ – Електронна обчислювальна машина

КПО – Коефіцієнт природної освітленості

ПЕОМ – Персональні електронні обчислювальні машини

ВСТУП

Основою людського існування є трудова діяльність, однак збереження життя, здоров'я і працездатності працівників у процесі виробництва вимагає знання охорони праці та вміння працювати з дотриманням правил безпеки, в яких сконцентрований досвід попередніх поколінь, сформульовані заходи профілактики професійних ризиків. Важливим є і те, що вивчаючи минулий досвід інших, працівники отримують можливість захистити себе в майбутньому. Основна мета навчання з охорони праці – попередження виробничого травматизму та професійної захворюваності.

На сьогоднішній день кожне підприємство використовує інформаційні технології, а саме працює за допомогою різних технічних пристроїв. Наприклад, персональний комп'ютер використовується для збору, обміну та аналізу інформації всередині підприємства та між іншими установами. Таке використання персонального комп'ютера дуже полегшує різні процеси на підприємстві. Але треба зазначити, що вплив комп'ютера несе не тільки позитивні наслідки, але й негативні. Використання комп'ютера загострило проблеми суспільного, а насамперед власного здоров'я, що вимагає удосконалення існуючих робочих місць та розробки нових, більш позитивних до здоров'я робочих місць та вимагає проведення профілактичних заходів до запобігання розвитку негативних наслідків впливу персонального комп'ютера на здоров'я користувача.

Досить багато проблем зі здоров'ям можуть виникнути через використання комп'ютера. Наприклад, проблеми із зором, проблеми зі спиною, також від комп'ютера надходить електромагнітне випромінювання.

Саме тому, у даному розділі слід розглянути питання гігієнічні вимоги умов праці користувачів ПК для запобігання проблем із здоров'ям, а також для підвищення працездатності.

1 ГІГІЄНІЧНІ ВИМОГИ ДО ОРГАНІЗАЦІЇ РОБОТИ З ВІЗУАЛЬНИМИ ДИСПЛЕЙНИМИ ТЕРМІНАЛАМИ

1.1 Вимоги до виробничих приміщень для експлуатації ВДТ ЕОМ та ПЕОМ

Згідно з «Державними санітарними правилами і нормами роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПІН 3.3.2.007-98.

Об'ємно-планувальні рішення будівель та приміщень для роботи з ВДТ ЕОМ і ПЕОМ мають відповідати вимогам цих Правил [6].

Розміщення робочих місць з ВДТ ЕОМ і ПЕОМ у підвальних приміщеннях, на цокольних поверхах заборонено.

Площа на одне робоче місце має становити не менше ніж $6,0 \text{ м}^2$, а об'єм не менше ніж $20,0 \text{ м}^3$.

Приміщення для роботи з ВДТ повинні мати природне та штучне освітлення відповідно до СНиП II-4-79.

Природне освітлення має здійснюватись через світлові прорізи, орієнтовані переважно на північ чи північний схід і забезпечувати коефіцієнт природною освітленості (КПО) не нижче ніж 1,5%. Розраховується КПО за методикою, викладеною в СНиП II-4-79.

Виробничі приміщення для роботи з ВДТ (операторські, диспетчерські) не повинні межувати з приміщеннями, в яких рівні шуму і вібрації перевищують допустимі значення (виробничі цехи, майстерні тощо) за СН 3223-85, СН 3044-84, ГР 2411-81, ГОСТ 12.1.003-83.

Звукоізоляція огорожувальних конструкцій приміщень з ВДТ має забезпечувати параметри шуму, що відповідають вимогам СН 3223-85, ГОСТ 12.1.003-83, ГОСТ 12.1.012-90.

Приміщення для роботи з ВДТ мають бути обладнані системами опалення, кондиціонування повітря, або припливно-витяжною вентиляцією відповідно до

СНиП 2.04.05-91. Нормовані параметри мікроклімату, іонного складу повітря, вмісту шкідливих речовин мають відповідати вимогам СН 4088-86, СН 2152-80, ГОСТ 12.1.005-88, ГОСТ 12.1.007-76.

Віконні прорізи приміщень для роботи з ВДТ мають бути обладнані регульованими пристроями (жалюзі, завіски, зовнішні козирки).

Для внутрішнього оздоблення приміщень з ВДТ слід використовувати дифузно-відбивні матеріали з коефіцієнтами відбиття для стелі 0,7-0,8, для стін 0,5-0,6.

Покриття підлоги повинне бути матовим з коефіцієнтом відбиття 0,3-0,5. Поверхня підлоги має бути рівною, неслизькою, з антистатичними властивостями.

Забороняється для оздоблення інтер'єру приміщень ВДТ застосовувати полімерні матеріали (деревинно-стружкові плити, шпалери, що миються, рулонні синтетичні матеріали, шаруватий паперовий пластик тощо), що виділяють у повітря шкідливі хімічні речовини.

Полімерні матеріали для внутрішнього оздоблення приміщень з ВДТ можуть бути використані при наявності дозволу органів та установ державної санітарно-епідеміологічної служби.

Виробничі приміщення можуть обладнуватись шафами для зберігання документів, магнітних дисків, полицями, стелажми, тумбами тощо з урахуванням вимог до площі приміщень.

У приміщеннях з ВДТ слід щоденно робити вологе прибирання.

Приміщення з ВДТ мають бути оснащені аптечками першої медичної допомоги.

При приміщеннях з ВДТ мають бути обладнані побутові приміщення для відпочинку під час роботи, кімната психологічного розвантаження. В кімнаті психологічного розвантаження слід передбачити встановлення пристроїв для приготування й роздачі тонізуючих напоїв, а також місця для занять фізичною культурою (СНиП 2.09.04. -87).

2 ГІГІЄНІЧНІ ВИМОГИ ДО ПАРАМЕТРІВ ВИРОБНИЧОГО СЕРЕДОВИЩА ПРИМІЩЕНЬ З ВДТ ЕОМ ТА ПЕОМ

2.1 Мікроклімат

У виробничих приміщеннях на робочих місцях з ВДТ мають забезпечуватись оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря (ГОСТ 12.1.005-88, СН 4088-86).

Рівні позитивних і негативних іонів у повітрі приміщень з ВДТ мають відповідати санітарно-гігієнічним нормам N 2152-80 [6].

2.1 Освітлення

Освітлення також впливає на користувача і має негативний вплив якщо воно неправильне.

Вимоги до природного освітлення викладено в п.п. 2.4,2.5.

Штучне освітлення в приміщеннях з робочими місцями, обладнаними ВДТ ЕОМ та ПЕОМ, має здійснюватись системою загального рівномірного освітлення. У виробничих та адміністративно-громадських приміщеннях, у разі переважної роботи з документами, допускається застосування системи комбінованого освітлення (крім системи загального освітлення додатково встановлюються світильники місцевого освітлення).

Зазначення освітлення освітленості на поверхні робочого столу в зоні розміщення документів має становити 300-500 лк. Якщо ці значення освітленості неможливо забезпечити системою загального освітлення, допускається використовувати місцева освітлення. При цьому світильники місцевого освітлення слід встановлювати таким чином, щоб не створювати бліків на поверхні екрана, а освітленість екрана має не перевищувати 300 лк.

Як джерела світла в разі штучного освітлення мають застосовуватись переважно люмінесцентні лампи типу ЛБ. У разі влаштування відбитого освітлення у виробничих та адміністративно-громадських приміщеннях допускається застосування металогалогенних ламп потужністю 250 Вт.

Допускається застосування ламп розжарювання у світильниках місцевого освітлення.

Система загального освітлення має становити суцільні або преривчасті лінії світильників, розташовані збоку від робочих місць (переважно ліворуч), паралельно лінії зору працюючих.

Допускається використання світильників таких класів світорозподілу

1. прямого світла - П;
2. переважно прямого світла - Н;
3. переважно відбитого світла - В.

Для загального освітлення слід застосовувати світильники серії ЛПО 3б із дзеркальними ґратами, укомплектовані високочастотними пускорегулювальними апаратами (ВЧ ПРА). Допускається застосовувати світильники цієї серії без ВЧ ПРА тільки в модифікації "Кососвітло". Застосування світильників без розсіювачів та екрануючих ґрат заборонено.

Яскравість світильників загального освітлення в зоні кутів випромінювання від 50 до 90 град. з вертикаллю в повздовжній та поперечній площинах має становити не більше ніж 200 кд/м², захисний кут світильників - не менше ніж 40 град.

Світильники місцевого освітлення повинні мати просвічуючий відбивач із захисним кутом, не меншим ніж 40 град.

Слід передбачити обмеження прямої блискості від джерел природного та штучного освітлення. При цьому яскравість світлих поверхонь (вікна, джерела штучного освітлення), що розташовані в полі зору повинна бути не більше ніж 200 кд/м².

Необхідно обмежувати відбиту блискість на робочих поверхнях відносно джерел природного і штучного освітлення. При цьому яскравість бліків на екрані ВДТ має не перевищувати 40 кд/м², а яскравість стелі в разі застосування системи відбитого освітлення 200 кд/м².

Показник осліпленості у разі використання джерел загального штучного освітлення у виробничих приміщеннях має не перевищувати 20, а показник дискомфорту в адміністративно-громадських приміщеннях має бути не більше за 40.

Необхідно обмежувати нерівномірність розподілу яскравості в полі зору працюючих з ВДТ. При цьому співвідношення яскравостей робочих поверхонь має бути не більшим ніж 3:1, а співвідношення яскравостей робочих поверхонь та поверхонь стін, обладнання тощо - 5:1.

Коефіцієнт запасу (К куб.) для освітлювальних установок загального освітлення має дорівнювати 1,4.

Коефіцієнт пульсації має не перевищувати 5%, що забезпечується застосуванням газорозрядних ламп у світильниках загального та місцевого освітлення з ВЧ ПРА для світильників будь-яких типів. Якщо не має світильників з ВЧ ПРА, то лампи багатолампових світильників або світильники загального освітлення, розташовані поруч, слід вмикати на різні фази трьохфазної мережі.

Для забезпечення нормованих значень освітленості у приміщеннях з ВДТ ЕОМ та ПЕОМ слід чистити шибки і світильники принаймні двічі на рік і вчасно замінювати лампи, що перегоріли.

2.3 Шум і вібрація

Відомо, що шум несприятливо діє на слуховий аналізатор та інші органи та системи організму людини. Визначальне значення щодо такої дії має інтенсивність шуму, його частотний склад, тривалість щоденного впливу, індивідуальні особливості людини, а також специфіка виробничої діяльності.

Рівні звукового тиску в октавних смугах частот, рівні звуку та еквівалентні рівні звуку на робочих місцях, обладнаних ВДТ ЕОМ і ПЕОМ, мають відповідати вимогам СН 3223-85, ГОСТ 12.1.003-83, ГР 2411-81.

Устаткування, що становить джерело шуму (АЦП, принтери тощо), слід розташовувати поза приміщенням для роботи ВДТ ЕОМ і ПЕОМ.

Для забезпечення допустимих рівнів шуму на робочих місцях слід застосовувати засоби звукопоглинання, вибір яких має обґрунтовуватись спеціальними інженерно-акустичними розрахунками.

Під час виконання робіт з ВДТ ЕОМ і ПЕОМ у виробничих приміщеннях значення характеристик вібрації на робочих місцях мають не перевищувати допустимі відповідно до СН 3044-84, ГОСТ 12.1.012-90.

2.4 Неіонізуючі електромагнітні випромінювання

Значення напруженості електростатичного поля на робочих місцях з ВДТ (як у зоні екрана дисплея, так і на поверхнях обладнання, клавіатури, друкувального пристрою) мають не перевищувати гранично допустимих за ГОСТ 12.1.045-84, СН 1757-77.

Значення напруженості електромагнітних полів на робочих місцях з ВДТ мають відповідати нормативним значенням (ГДР N 3206-85, ГДР N 4131-86, СН N 5802-91, ГОСТ 12.1.006-84).

Інтенсивність потоків інфрачервоного випромінювання має не перевищувати допустимих значень відповідно до СН 4088-86, ГОСТ 12.1.005-88.

Інтенсивність потоків ультрафіолетового випромінювання має не перевищувати допустимих значень відповідно до СН 4557-88.

2.5 Іонізуючі електромагнітні випромінювання

Іонізуючі електромагнітні випромінювання на відстані 0,05 м від екрана до корпусу відеотермінала при будь-яких положеннях регулювальних пристроїв не повинна перевищувати $7,74 \times 10^{-12}$ А/кг, що відповідає еквівалентній дозі 0,1 мбер/год (100 мкР/год) НРБУ N 58.

3 ГІГІЄНІЧНІ ВИМОГИ ДО ОРГАНІЗАЦІЇ І ОБЛАДНАННЯ РОБОЧИХ МІСЦЬ З ВДТ ЕОМ І ПЕОМ

Гігієна також має певний вплив, тому слід дотримуватись правил.

Обладнання і організація робочого місця працюючих з ВДТ ЕОМ і ПЕОМ мають забезпечувати відповідність конструкції всіх елементів робочого місця та їх взаємного, розташування ергономічним вимогам з урахуванням характеру і особливостей трудової діяльності (ГОСТ 12.2.032-78, ГОСТ 22.269-76, ГОСТ 21.889-76).

Конструкція робочого місця користувача ЕОМ і ПЕОМ з ВДТ має забезпечити підтримання оптимальної робочої пози.

Робочі місця з ВДТ слід так розташовувати відносно світових прорізів, щоб природне світло падало збоку переважно зліва.

При розміщенні робочих столів з ВДТ слід дотримувати такі відстані між бічними поверхнями ВДТ 1,2 м, відстань від тильної поверхні одного ВДТ до екрана іншого ВДТ - 2,5 м.

Конструкція робочого столу має відповідати сучасним вимогам ергономіки і забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання (дисплея, клавіатури, принтера) і документів.

Висота робочої поверхні робочого столу з ВДТ має регулюватися в межах 680...800 мм, а ширина і глибина - забезпечувати можливість виконання операцій у зоні досяжності моторного поля (рекомендовані розміри: 600...1400 мм, глибина - 800..1000 мм).

Робочий стіл повинен мати простір для ніг заввишки не менше ніж 600 мм, завширшки не менше ніж 500 мм, завглибшки (на рівні колін) не менше ніж 450 мм, на рівні простягнутої ноги -ніж 650 мм.

Робочий стілець має бути підйомно-поворотним, регульованим за висотою, з кутом і нахилу сидіння та спинки і за відстанню від спинки до переднього краю сидіння поверхня сидіння має бути плоскою, передній край - заокругленим.

Регулювання за кожним із параметрів має здійснюватися незалежно, легко і надійно фіксуватися. Шаг регулювання елементів стільця має становити: для лінійних розмірів -15...20 мм, для кутових 2...5 град. Зусилля регулювання має не перевищувати 20 Н.

Висота поверхні сидіння має регулюватися в межах 400...500 мм, а ширина і глибина становити не менше ніж 400 мм. Кут нахилу сидіння - до 15 град. вперед і до 5 град. назад.

Висота спинки стільця має становити (300+/-20) мм, ширина - не менше ніж 380 мм, радіус кривизни горизонтальної площини - 400 мм. Кут нахилу спинки має регулюватися в межах 1...30 град. від вертикального положення. Відстань від спинки до переднього краю сидіння має регулюватися в межах 260...400 мм.

Для зниження статичного напруження м'язів верхніх кінцівок слід використовувати стаціонарні або змінні підлокітники завдовжки не менше ніж 250 мм, завширшки 50...70 мм, що регулюються за висотою над сидінням у межах 230...260 мм і відстанню між підлокітниками в межах 350...500 мм.

Поверхність сидіння і спинки стільця має бути напівм'якою з нековзним, повітронепроникним покриттям, що легко чиститься і не електризується.

Робоче місце має бути обладнане підставкою для ніг завширшки не менше ніж 300 мм, завглибшки не менше ніж 400 мм, що регулюється за висотою в межах до 150 мм і за кутом нахилу опорної поверхні підставки до 20 град. Підставка повинна мати рифлену поверхню і бортик по передньому краю заввишки 10 мм.

Екран ВДТ має розташовуватися на оптимальній відстані від очей користувача, що становить 600...700 мм, але не ближче ніж за 600 мм з урахуванням розміру літерно-цифрових знаків і символів.

Розташування екрана ВДТ має забезпечувати зручність зорового спостереження у вертикальній площині під кутом +30 град. до нормальної лінії погляду працюючого.

Клавіатуру слід розташовувати на поверхні столу на відстані 100...300 мм від краю, звернутого до працюючого. У конструкції клавіатури має передбачатися опорний пристрій (виготовлений із матеріалу з високим коефіцієнтом тертя, що перешкоджає мимовольному її зсуву), який дає змогу змінювати кут нахилу поверхні клавіатури у межах 5...15 град. Висота середнього рядка клавіш має не перевищувати 30 мм. Поверхня клавіатури має бути матовою з коефіцієнтом відбиття 0,4.

Розташування пристрою введення - виведення інформації має забезпечувати добру видимість екрана ВДТ, зручність ручного керування в зоні досяжності моторного поля і за висотою - 900...1300 мм, за шириною 400...500 мм.

Робоче місце з ВДТ слід обладнати попітром для документів, що легко переміщуються.

Для забезпечення захисту і досягнення нормованих рівнів комп'ютерних випромінювань необхідно застосування приєкранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, що пройшли випробування в акредитованих лабораторіях і мають щорічний гігієнічний сертифікат.

При оснащеності робочого місця з ВДТ лазерним принтером параметри лазерного випромінювання повинні відповідати вимогам СанПіН N 5804-91.

4 ВИМОГИ ДО РЕЖИМІВ ПРАЦІ І ВІДПОЧИНКУ ПРИ РОБОТІ З ВДТ ЕОМ І ПЕОМ

При організації праці, що пов'язана з використанням ВДТ ЕОМ і ПЕОМ, для збереження здоров'я працюючих, запобігання професійним захворювання і підтримки працездатності слід передбачити внутрішньозмінні регламентовані перерви для відпочинку.

Внутрішньозмінні режими праці і відпочинку мають передбачати додаткові нетривалі перерви в періоди, що передують появі об'єктивних і суб'єктивних ознак втомлення і зниження працездатності.

При виконанні протягом дня робіт, що належать до різних видів трудової діяльності, за основну роботу з ВДТ ЕОМ і ПЕОМ слід вважати таку, що займає не менше 50% часу впродовж робочої зміни мають передбачатися:

- 1. перерви для відпочинку і вживання їжі (обідні перерви);**
- 2. перерви для відпочинку і особистих потреб (згідно з трудовими нормами);**
- 3. додаткові перерви, що вводяться для окремих професій з урахуванням особливостей трудової діяльності.**

Тривалість обідньої перерви визначається чинним законодавством про працю і Правилами внутрішнього трудового розпорядку підприємства (Організації, установи).

Внутрішньозмінні режими праці і відпочинку при роботі з ВДТ ЕОМ і ПЕОМ розроблено з урахуванням характеру трудової діяльності, напруженості і важкості праці диференційовано для кожної професії.

За характером трудової діяльності виділено три професійні групи згідно з діючим класифікатором професій (ДК 003-95 (v0257217-95, va257217-95, vb257217-95) і Зміна N 1 до ДК 003-95 (v0314217-98)):

1. розробники програм (інженери-програмісти) - виконують роботу переважно з відеотерміналом та документацією при необхідності і інтенсивного обміну інформацією з ЕОМ і високою частиною прийняття рішень. Робота характеризується інтенсивною

- розумовою творчою працею з підвищеним напруженням зору, концентрацією уваги на фоні нервово-емоційного напруження, вимушеною робочою позою, загальною гіподинамією, періодичним навантаженням на кисті верхніх кінцівок. Робота виконується в режимі діалогу з ЕОМ у вільному темпі з періодичним пошуком помилок в умовах дефіциту часу;
2. оператори електронно-обчислювальних машин - виконують роботу яка пов'язана з обліком інформації одержаної з ВДТ за попереднім запитом, або тієї, що надходить з нього, супроводжується перервами різної тривалості, пов'язана з виконанням іншої роботи і характеризується як робота з напруженням зору, невеликими фізичними зусиллями, нервовим напруженням середнього ступеня та виконується у вільному темпі;
 3. оператор комп'ютерного набору - виконує одноманітні за характером роботи з документацією та клавіатурою і нечастими нетривалими переключеннями погляду на екран дисплея, з введенням даних з високою швидкістю, робота характеризується як фізична праця з підвищеним навантаженням на кисті верхніх кінцівок на фоні загальної гіподинамії з напруженням зору (фіксація зору переважно на документи), нервово-емоційним напруженням.

Встановлюються такі внутрішньозмінні режими праці та відпочинку при роботі з ЕОМ при 8-годинній денній робочій зміні в залежності від характеру праці:

1. для розробників програм із застосуванням ЕОМ, слід призначати регламентовану перерву для відпочинку тривалістю 15 хвилин через кожну годину роботи за ВДТ;
2. для операторів із застосування ЕОМ, слід призначати регламентовані перерви для відпочинку тривалістю 15 хвилин через кожні дві години;
3. для операторів комп'ютерного набору слід призначати регламентовані перерви для відпочинку тривалістю 10 хвилин після кожною години роботи за ВДТ.

У всіх випадках, коли виробничі обставини не дозволяють застосувати регламентовані перерви, тривалість безперервної роботи з ВДТ не повинна перевищувати 4 години.

При 12-годинній робочій зміні регламентовані перерви повинні встановлюватися в перші 8 годин роботи аналогічно перервам при 8-годинній робочій зміні, а протягом останніх 4-х годин роботи, незалежно від характеру трудової діяльності, через кожну годину тривалістю 15 хвилин.

З метою зменшення негативного впливу монотонності є доцільним застосовувати чергування операцій усвідомленого тексту і числових даних (зміна змісту роботи). Чередування вводу даних та редагування текстів.

Для зниження нервово-емоційного напруження, втомлення зорового аналізатору, поліпшення мозкового кровообігу, подолання несприятливих наслідків гіподинамії, запобігання втоми доцільні деякі перерви використовувати для виконання комплексу вправ.

В окремих випадках - при хронічних скаргах працюючих з ВДТ ЕОМ і ПЕОМ на зореве втомлення незважаючи на дотримання санітарно-гігієнічних вимог до режимів праці і відпочинку, а також застосування засобів локального захисту очей - допускаються індивідуальних підхід до обмеження часу робіт з ВДТ, зміни характеру праці, чергування з іншими видами діяльності, не пов'язаними з ВДТ.

Активний відпочинок має полягати у виконанні комплексу гімнастичних вправ, спрямованих на зняття нервового напруження, м'язове розслаблення, відновлення функцій фізіологічних систем, що порушуються протягом трудового процесу, зняття втоми очей, поліпшення мозкового кровообігу і працездатності.

ВИСНОВКИ

Під час виконання спеціальної частини з охорони праці було проаналізовано умови праці людини на підприємствах та установах, а також роботу при користуванні електронними пристроями.

Правила, які прописані у нормативних документах є актуальними на сьогодні через те, що майже в кожному підприємстві є працівники, які працюють з електронними пристроями. Під час роботи за персональним комп'ютером людина швидко втомлюється та від цього страждає опорно-руховий, зоровий апарат, нервова система и в окремих випадках психічне здоров'я.

На сьогодні існує багато шкідливих умов, які погіршують процес трудової діяльності людини, такі як: шум, вібрація та інші умови.

Вимоги та нормативи, щодо охорони праці мають дотримуватися на підприємствах під час трудової діяльності людини. Дотримання поставлених вимог до працівників та власників підприємств дозволить мінімізувати шкідливі наслідки, які мають вплив на здоров'я людини.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. ГОСТ 12.1.005-88 Система стандартов безопасности труда (ССБТ). Общие санитарно-гигиенические требования к воздуху рабочей зоны (с Изменением N 1) от 29 сентября 1988 - docs.cntd.ru. URL: <https://docs.cntd.ru/document/1200003608> (accessed 14/06/2022).
2. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПІН 3.3.2.007-98. URL: <https://mozdocs.kiev.ua/view.php?id=2445> (accessed 14/06/2022).
3. ДНАОП 3044-84. СН 3044-84 (ДНАОП 0.03-3.12-84) Санітарні норми вібрації робочих місць (43248). URL: https://dnaop.com/html/43248/doc - ДНАОП_3044-84 (accessed 14/06/2022).
4. ДСН 3.3.6.042-99. Санітарні норми мікроклімату виробничих приміщень (34094). URL: https://dnaop.com/html/34094/doc -ДСН_3.3.6.042-99 (accessed 14/06/2022).
5. Про затвердження Вимог щодо безп... | від 14.02.2018 № 207. URL: <https://zakon.rada.gov.ua/laws/show/z0508-18#Text> (accessed 14/06/2022).
6. Про затвердження Правил охорони п... | від 10.02.1999 № 21. URL: <https://zakon.rada.gov.ua/laws/show/z0382-99#Text> (accessed 14/06/2022).
7. СНиП 2.09.04-87*. Административные и бытовые здания (54074). URL: https://dnaop.com/html/54074/doc -СНиП_2.09.04-87_ (accessed 14/06/2022).
8. СНиП II-4-79*. СНиП II-4-79*. Естественное и искусственное освещение (отменен -приказ N168 от 15 мая 2006 г.) (45036). URL: https://dnaop.com/html/45036/doc- СНиП_II-4-79_ (accessed 14/06/2022).