

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

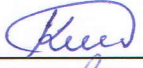
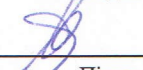
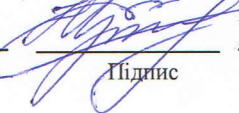
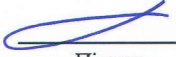
КВАЛІФІКАЦІЙНА РОБОТА

«Вебзастосунок для продажу кондитерських виробів»

Назва теми

Рівень вищої освіти перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного
забезпечення»

Шифр КвРПЗ.2201112.01.17.ПЗ

Виконав студент IV курсу групи ПЗ-22-1  Кіріл ТРЕМБАЧ
Підпис Ім'я, прізвище
Керівник канд. техн. наук, доцент  Оксана ЯШИНА
Науковий ступінь, звання Підпис Ім'я, прізвище
Нормоконтролер канд. пед. наук, доцент  Наталія ПРАВОРСЬКА
Науковий ступінь, звання Підпис Ім'я, прізвище
До захисту допускаю:
Завідувач кафедри інженерії
програмного забезпечення  Леонід БЕДРАТЮК
Підпис Ім'я, прізвище

2 червня 2026 р.

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Інженерії програмного забезпечення

Рівень вищої освіти Перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Л. П. Бедратюк

02 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Трембачу Кірілу Олеговичу

Прізвище, ім'я, по батькові студента

1. Тема роботи «Вебзастосунок для продажу кондитерських виробів»

Керівник роботи Яшина Оксана Миколаївна, канд. техн. наук, доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. №7.

2. Строк подання студентом роботи на кафедру 01.06.26 р.

3. Вихідні дані до роботи Методичні матеріали до кваліфікаційної роботи

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Дослідження предметної області та постановка задачі, проектування програмного забезпечення, програмна реалізація, тестування онлайн-ресурсу

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Три креслення:

1. Діаграма варіантів використання

2. Схема бази даних

3. Архітектурне рішення

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Праворська Н. І., доцент кафедри ІІЗ	05.05.26	08.05.26
Антиплагіат	Форкун Ю. В., доцент кафедри ІІЗ	05.05.2026	25.05.26

7. Дата видачі завдання «02» січня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

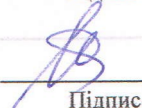
Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1 Ознайомлення з тематикою кваліфікаційної роботи (КвР), визначення та узгодження індивідуальних тем КвР	01.12 – 31.12.2025	
2 Збір матеріалу за темою КвР; дослідження предметної області, в якій планується використання програмного забезпечення (ІІЗ), визначення задач та вимог, розробка технічного завдання	02.01 – 31.01.2026	
3 Проектування програмного забезпечення	01.02 – 28.02.2026	
4 Програмна реалізація з використанням відповідних засобів розробки	01.03 – 10.04.2026	
5 Тестування програмного забезпечення	11.04 – 30.04.2026	
6 Написання вступу, загальних висновків, оформлення переліку джерел посилання та додатків. Оформлення пояснювальної записки КвР згідно вимог	01.05 – 25.05.2026	
7 Попередній захист КвР	травень 2026 (згідно графіка)	
8 Перевірка КвР на плагіат, нормоконтроль, отримання відгуків, рецензій та інших супровідних документів. Брошування (зшиття) пояснювальної записки.	26.05 – 30.05.2026	
9 Здача КвР на кафедру; підготовка КвР для розміщення у репозиторії ХНУ; підготовка до захисту та захист КвР	з 01.06.2026	

Студент


Підпис

Кіріл ТРЕМБАЧ
Ініціали, прізвище

Керівник роботи


Підпис

Оксана ЯШИНА
Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Вебзастосунок для продажу кондитерських виробів».

Автор роботи: Трембач Кіріл Олегович.

Керівник роботи: Яшина Оксана Миколаївна.

Пояснювальна записка: 81 с., 27 рис., 12 табл., 2 дод., 30 джерел.

Графічна частина: 3 креслення у форматі А3.

ВЕБЗАСТОСУНОК, КОНДИТЕРСЬКІ ВИРОБИ, ОНЛАЙН-ОПЛАТА,
ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ПРОГРАМНИЙ ПРОДУКТ.

Метою кваліфікаційної роботи є розробка вебзастосунку для продажу кондитерських виробів, що містить функціональні можливості для купівлі солодощів, користуючись мережею Інтернет.

У ході виконання кваліфікаційної роботи проаналізовано предметну область, а саме реалізацію кондитерських виробів, встановлено мету із відповідними завданнями. Також встановлено вимоги до даного вебзастосунку, здійснено проектування архітектури, бази даних, інтерфейсу користувача.

У результаті отримано потрібний вебзастосунок для продажу кондитерських виробів, на якому користувач (потенційний клієнт) може шукати, сортувати, оформляти замовлення та купувати кондитерські вироби.

28 травня
Дата


Підпис

ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	КвРІПЗ.2201112.01.17.ПЗ	Пояснювальна записка	82		
2	A4		Завдання на кваліфікаційну роботу	2		
3	A4		Анотація	1		
			<u>Графічні документи</u>			
4	A3	КвРІПЗ.2201112.01.17.ПЗ	Діаграма варіантів використання	1		
5	A3	КвРІПЗ.2201112.01.17.ПЗ	Схема бази даних	1		
6	A3	КвРІПЗ.2201112.01.17.ПЗ	Архітектура	1		





КвРІПЗ.2201112.01.17.ПЗ

Змн.	Арк.	№ докум.	Підпис	Дата				
Виконав		Трембач К.О.	<i>Керд</i>	29.05	Вебзастосунок для продажу кондитерських виробів	Літ.	Арк.	Аркушів
Керівник		Яшина О.М.	<i>Я</i>	23.05			5	65
Н. контр.		Праворська Н.І.	<i>Н.І. Праворська</i>	23.05	Відомість документів	ХНУ, ІПЗ-22-1		
Зав. каф.		Бедратюк Л.П.	<i>Л.П. Бедратюк</i>	23.05				

ЗМІСТ

Вступ	7
1 Дослідження предметної області та постановка задачі.....	9
1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей	9
1.2 Аналіз наявного програмно-технічного забезпечення предметної області	11
1.3 Визначення функціональних та нефункціональних вимог до програмного забезпечення.	13
1.4 Висновки. Постановка задачі.....	20
2 Проектування вебзастосунку	21
2.1 Проектування архітектури та структури вебзастосунку	21
2.2 Проектування логічної моделі бази даних вебзастосунку	24
2.3 Проектування інтерфейсу користувача вебзастосунку	25
2.4 Аналіз та вибір технологій і методів реалізації вебзастосунку	28
3 Програмна реалізація та тестування вебзастосунку	35
3.1 Реалізація бази даних вебзастосунку	35
3.2 Реалізація модулів вебзастосунку	43
3.3 Інструкція користувача вебзастосунку	51
3.4 Тестування вебзастосунку.....	54
Висновки.....	60
Перелік джерел посилання.....	62
Додаток А Презентаційні матеріали	65
Додаток Б Технічне завдання.....	73

КвРІПЗ.2201112.01.17.ПЗ

Змн.	Арк.	№ докум.	Підпис	Дата				
Виконав		Трембач К.О.		29.05	Вебзастосунок для продажу кондитерських виробів	Літ.	Арк.	Аркушів
Керівник		Яшина О.М.		29.05			6	65
Н. контр.		Праворська Н.І		29.05	Відомість документів	ХНУ, ІПЗ-22-1		
Зав. каф.		Бедратюк Л.П.		29.05				

ВСТУП

Створення вебзастосунку для кондитерської галузі це не просто крок назустріч трендам, а базовий інструмент виживання та масштабування. Солодкий бізнес тримається на візуальній естетиці та емоційних покупках, а цифровий простір є найкращим майданчиком для їхньої стимуляції.

Сучасний споживач звик купувати «очима». Якісний вебзастосунок дозволяє продемонструвати текстуру кремів, деталізацію декору та розрізи десертів у високій роздільній здатності, що фізично важко реалізувати на обмеженій площі вітрини в кав'ярні. Це створює ефект присутності та викликає негайне бажання спробувати продукт. Окрім того, персоналізація замовлень - наприклад, конструктор тортів, де клієнт сам обирає начинку, вагу та напис - стає можливою лише завдяки автоматизованим інтерфейсам, які знімають навантаження з менеджерів.

Інтеграція онлайн-оплати радикально змінює фінансову модель кондитерської. По-перше, вона мінімізує ризик «незабраних» замовлень: коли клієнт оплачує десерт одразу, він з набагато більшою ймовірністю його забере, що критично для продуктів із коротким терміном придатності. По-друге, це пришвидшує процес обробки замовлення, адже кошти миттєво зараховуються на рахунок, а система автоматично формує чек та передає завдання на кухню.

У світі, де час є найціннішим ресурсом, можливість зробити замовлення в три кліки через смартфон стає вирішальним фактором при виборі кондитерської. Якщо конкурент змушує клієнта телефонувати або писати в Direct для уточнення ціни, власник бізнесу пропонує зручний кошик із миттєвою оплатою Apple Pay або Google Pay – то він виграє битву за лояльність. До того ж вебзастосунок дозволяє впровадити складні системи бонусів та накопичувальних знижок, які стимулюють покупця повертатися саме до вас.

					КвРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ док.ум.	Підпис	Дата		7

Актуальність такої розробки також підкріплюється розвитком служб доставки. Вебзастосунок із чітким API дозволяє легко інтегруватися з логістичними сервісами, забезпечуючи клієнту шлях від хочу цей еклер до кур'єр уже під дверима за лічені хвилини.

Мета роботи - розробка вебзастосунку для продажу кондитерських виробів, що містить функціональні можливості для купівлі солодощів, користуючись мережею Інтернет.

Для того, щоб була досягнута мета потрібно здійснити розв'язок наступних завдань:

- здійснити аналіз ринку реалізації кондитерських виробів онлайн;
- проаналізувати існуючі програмні рішення, що допомагають вести онлайн-бізнес з продажу кондитерських виробів;
- встановити вимоги до розробки програмного забезпечення;
- здійснити розробку технічного завдання;
- проаналізувати та обрати тип архітектури, а також проектування бази даних;
- провести реалізацію вебзастосунку для продажу кондитерських виробів;
- здійснити тестування вебзастосунку для реалізації кондитерських виробів.

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		8

1 ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

Кондитерська галузь сьогодні - це вже не просто випікання тортів, а повноцінний сегмент FoodTech, де перемагає той, хто пропонує найкращий цифровий клієнтський досвід. Інтернет-торгівля кондитерськими виробами в Україні набуває все більшого значення.

Предметна область охоплює процеси виробництва, складського обліку, логістики та маркетингу десертів, що має кілька специфічних рис:

- гіпер-персоналізація, оскільки клієнти більше не хочуть «просто торт». Вони хочуть конструктор, де можна обрати все - від заміни цукру до 3D-друку їстівного декору.

- омніканальність, тобто бізнес має бути всюди (Instagram, TikTok, вебсайт), але всі замовлення повинні стікатися в одну «розумну» систему.

- прозорість складу, оскільки кондитерські вироби мають дуже короткий термін придатності, система повинна точно знати, скільки вершків чи полуниці залишилося, щоб не прийняти замовлення, яке неможливо виконати.

Якщо раніше сайт був «візитівкою», то зараз це головний операційний вузол. Ось основні фактори актуальності:

1. Економічна стабільність та передоплата.

У кондитерській справі «відмова» від замовлення - це прямі збитки (сировина + час майстра). Вебзастосунок з інтегрованою оплатою (Monoray/Stripe) переводить бізнес на модель 100% гарантії. Клієнт, який сплатив, не забуде забрати свій десерт.

2. Звільнення від залежності від месенджерів.

Адміністратори кондитерських витрачають до 60% часу на уточнення начинки, дати та вартості в Instagram Direct. Автоматизований конструктор у

					КвРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		9

вебзастосунку робить це за секунди, дозволяючи персоналу фокусуватися на сервісі, а не на копіюванні цін.

3. Незалежність від платформ.

Залежність лише від соцмереж (Instagram/Facebook) - це ризик. Зміна алгоритмів чи блокування акаунта може зупинити продажі. Власний вебзастосунок на базі надійної інфраструктури - це власний «цифровий актив», який належить тільки вам.

4. Робота з даними (Big Data для малого бізнесу)

Застосунок дозволяє бачити:

- які начинки найпопулярніші у конкретний день;
- хто постійні клієнти (з автоматичною пропозицією знижки на день народження;
- середній чек та ефективність маркетингових акцій.

Таблиця 1.1 - Порівняльний аналіз

Параметр	Без застосунку (Ручний режим)	З вебзастосунком (FoodTech)
Приєм замовлення	10-15 хв переписки/дзвінків	2 хв самостійно клієнтом
Помилки в замовленні	Високий ризик (людський фактор)	Мінімальний (клієнт сам обирає параметри)
Контроль фінансів	Складна звірка чеків та виписок	Автоматичні звіти в адмін-панелі
Масштабованість	Потребує більше менеджерів	Система обробляє 1000 замовлень так само легко, як 10

Розробка такого застосунку - це не просто «тренд», а необхідність для виживання бізнесу. Це перехід від хаотичного продажу через месенджери до

структурованого ІТ-продукту, який забезпечує безпеку (технічну та фінансову) та масштабованість.

1.2 Аналіз наявного програмно-технічного забезпечення предметної області

Найбільш популярні платформи, що займаються реалізацією кондитерських виробів можна розділити на декілька груп. Аналіз існуючих рішень у ніші e-commerce для кондитерських виробів дозволяє виділити три основні типи розробок: великі маркетплейси, мережеві вебсайти (бренди) та платформи-конструктори.

Ось детальний аналіз із зазначенням переваг та недоліків кожного підходу.

1. Великі продуктові маркетплейси (наприклад Zakaz.ua, Glovo, Сільпо).

Це універсальні платформи, де кондитерські вироби є лише однією з категорій.

Переваги:

- висока довіра, бо користувачі вже мають акаунти та прив'язані картки;
- потужна логістика через налагоджену систему кур'єрської доставки;
- технологічність через наявність мобільних застосунків, відстеження замовлення в реальному часі.

До недоліків можна віднести:

- відсутність кастомізації, наприклад, неможливо замовити напис на торті або змінити інгредієнти;
- стандартизація, бо фото товарів часто виглядають сухо, немає «крафтової» атмосфери.
- висока комісія, бо для виробника участь у таких платформах коштує значних відсотків від продажу.

					КвРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		11

2. Спеціалізовані брендові вебзастосунки (напр., Roshen, Milk Bar, Lviv Handmade Chocolate)

Це власні розробки конкретних брендів.

Переваги:

- унікальний UX/UI, тобто дизайн повністю передає атмосферу закладу (використання фірмових кольорів, шрифтів, якісного фуд-фото);
- програми лояльності через пряму інтеграцію із внутрішніми бонусними системами;
- гнучкість замовлення, тобто можливість вибрати начинку, додати листівку або святкову упаковку.

Недоліки:

- локальність, що зазвичай обмежено одним містом або мережею.
- вартість розробки, оскільки створення та підтримка власного застосунку потребує значних інвестицій.

3. Сайти на конструкторах та SaaS (напр., Shopify, Wix, Хорошоп)

Найпопулярніший варіант для малих крафтових кондитерських.

Переваги:

- швидкий старт, оскільки є готові шаблони для вітрин;
- інтеграції завдяки вже підключеним платіжним системам та сервіси доставки (наприклад, Нова Пошта).

Недоліки:

- обмежений функціонал, бо складно реалізувати складний конструктор тортів;
- шаблонність, бо сайти часто схожі один на одного, що заважає бренду виділитися.

Таблиця 1.2 - Порівняльна таблиця характеристик існуючих програмних рішень

					КвРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		12

Характеристика	Маркетплейси	Брендові застосунки	Конструктори (SaaS)
UX/UI Дизайн	Стандартний	Унікальний	Шаблонний
Кастомізація товару	Відсутня	Висока	Середня
Швидкість розробки	Миттєво (реєстрація)	Довго (3-6 міс)	Швидко (1-2 тижні)
Вартість володіння	Комісія з продажів	Висока (підтримка)	Абонплата

Аналізуючи конкурентів, можна виділити критичні точки успіху для розробки:

- візуальний фокус, оскільки кондитерські вироби купують «очима». Перевагою розроблюваного застосунку має стати велика кількість медіа-контенту, наприклад, відео-огляди тортів у розрізі, 360-фото);
- гнучкий календар, бо більшість існуючих систем погано працюють із попереднім замовленням на конкретну дату. Реалізація зручного вибору слотів доставки буде великим плюсом;
- мобільна адаптація, оскільки понад 80% покупок їжі здійснюються зі смартфонів, застосунок повинен працювати за принципом Mobile First.

1.3 Визначення функціональних та нефункціональних вимог до програмного забезпечення

Функціональні вимоги до системи – це функціональний складник застосунку, що має забезпечувати повний цикл взаємодії покупця з магазином. Передусім система повинна надавати зручний каталог товарів із можливістю фільтрації за категоріями (торти, тістечка, макарони), ціною та наявністю

специфічних інгредієнтів чи алергенів, таких як глютен або лактоза. Кожна картка десерту має містити детальний опис складу, калорійності, ваги та якісну галерею зображень.

Під час роботи з кошиком та оформлення замовлення користувачу необхідно надати змогу гнучко змінювати кількість позицій із миттєвим автоматичним перерахунком загальної вартості.

Важливою вимогою, зумовленою специфікою кондитерської сфери, є можливість додати кастомні побажання (наприклад, текст напису на торті), а також обов'язковий вибір конкретної дати та вузького часового слота доставки. Крім того, система має підтримувати реєстрацію та авторизацію користувачів для збереження історії їхніх покупок. Для адміністратора ж необхідно передбачити захищену панель керування, де він зможе редагувати асортимент, актуалізувати ціни та змінювати статуси поточних замовлень.

Нефункціональні вимоги відповідають за якість та безпеку, тобто визначають те, як саме працюватиме застосунок, ключовим пріоритетом є продуктивність та адаптивність. Інтерфейс має проєктуватися за строгим принципом Mobile-First, гарантуючи ідеальне відображення на смартфонах із дотриманням зручних зон натискання для сенсорних екранів. Час завантаження сторінок, попри велику кількість графічного контенту, має бути мінімальним, що досягається завдяки технології лінивого завантаження зображень (lazy loading).

З точки зору інформаційної безпеки, базовим стандартом є використання захищеного протоколу HTTPS на всіх етапах сесії. Паролі користувачів повинні зберігатися в базі даних виключно у захешованому вигляді, а платіжні реквізити взагалі не мають проходити через власний сервер застосунку — для цього використовуються токени платіжних систем. Також архітектура повинна бути надійною та масштабованою, щоб система не давала збоїв під час пікових навантажень у святкові дні (День святого Валентина, 8 Березня, Новий рік).

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		14

Для повноцінного та сучасного функціонування застосунк вимагає налаштування зв'язків із низкою сторонніх платформ. Задля забезпечення онлайн-оплат необхідна інтеграція з API надійних платіжних шлюзів, таких як LiqPay, MonoPay або WayForPay. Оскільки проєкт передбачає зберігання великого обсягу високоякісних фотографій їжі, їх доцільно винести на спеціалізовані хмарні сховища на кшталт Cloudinary або AWS S3, які також забезпечать автоматичну оптимізацію ваги файлів.

Нарешті, для комунікації з клієнтами та персоналом система повинна взаємодіяти із сервісами сповіщень - генерувати автоматичні листи на email покупця після успішної оплати та надсилати миттєві сповіщення в Telegram-бот адміністратора про появу нового замовлення.

1. Вимоги до інфраструктури та інструментарію.

Розробка повинна здійснюватися виключно на обладнанні, що відповідає стандартам матеріально-технічного забезпечення компанії. Це включає використання робочих станцій з оперативною пам'яттю від 16 ГБ та обов'язкову роботу з двома моніторами для підвищення точності верстки та кодування.

Програмний стек має базуватися на середовищі віртуалізації Docker для забезпечення ідентичності локального та серверного оточення. Весь код підлягає обов'язковій версійності через Git-репозиторій.

2. Технологічний процес та контроль якості.

Кожна нова функція (наприклад, кошик або форма оплати) вважається завершеною лише після проходження Code Review. Цей процес є частиною технологічного забезпечення розробки та спрямований на мінімізацію вразливостей у платіжних модулях.

Оскільки розробка вебзастосунку з онлайн-оплатою вимагає високої концентрації, робочий процес має суворо відповідати правилам техніки безпеки компанії.

Розробник зобов'язаний дотримуватися ергономічних норм, забезпечуючи перерви для відпочинку очей, що знижує ризик помилок у

					КвРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		15

критичних ділянках коду. Також заборонено використання несправного периферійного обладнання, що може призвести до порушення правил електробезпеки або пожежної безпеки в офісі. У разі виникнення будь-яких несправностей апаратної частини, робота має бути припинена до втручання системного адміністратора.

Основні функціональні модулі вебзастосунку:

1. Каталог виробів з підтримкою фільтрації за категоріями та високою якістю візуалізації.

2. Модуль оплати з інтеграцією API платіжних систем (Apple Pay, Google Pay, банківські картки).

3. Адмін-панель для керування замовленнями в режимі реального часу.

Задача вважається виконаною, якщо код пройшов усі тести в СІ-системі, отримав схвальні відгуки під час Code Review, а його розгортання не спричинило критичних навантажень на апаратну інфраструктуру бази практики.

Для вибору платіжного шлюзу ми порівнюємо два найпопулярніші сценарії: Stripe, що є міжнародним стандартом з найкращою документацією та Monopay, що є лідером українського ринку за швидкістю зарахування та зручністю для локального бізнесу. Вибір конкретної системи вплине на те, як системні адміністратори бази практики налаштовуватимуть серверне програмне забезпечення та брандмауери.

Таблиця 1.3 - Порівняльна характеристика платіжних систем

Характеристика	Stripe (Глобальний вибір)	Monopay від Monobank (Локальний вибір)
Складність інтеграції	Потребує високої кваліфікації розробника, але має ідеальні SDK.	Дуже проста інтеграція, адаптована під українське законодавство та ПРРО.

Продовження таблиці 1.3:

Методи оплати	Apple Pay, Google Pay, картки будь-яких банків світу, підписки.	Apple Pay, Google Pay, QR-коди, оплата частинами, моно-чекауті.
Комісія	Вища (зазвичай від 2.9% + фіксована сума).	Конкурентна (близько 1.3-1.5%) для українського бізнесу.
Вимоги до сервера	Потребує налаштування Webhooks для відстеження статусів транзакцій.	Мінімальні вимоги, стабільна робота через прості HTTP-запити.

Для успішної реалізації модуля оплати розробник повинен забезпечити виконання наступних технічних стандартів. Програмне забезпечення має підтримувати асинхронну обробку запитів, щоб користувач не чекав відповіді банку на «заблокованій» сторінці.

Основним критерієм безпеки є реалізація Webhooks - спеціальних сповіщень від платіжної системи до вашого сервера. Це дозволяє вебзастосунку автоматично змінювати статус замовлення на «Оплачено» навіть у випадку, якщо клієнт випадково закрити вкладку браузера після транзакції. Таке налаштування потребує від системних адміністраторів відкриття відповідних портів у мережевій інфраструктурі бази практики, що має бути зроблено з дотриманням протоколів кібербезпеки.

Оскільки робота відбувається з фінансами, розробник зобов'язаний дотримуватися принципів мінімізації даних. Вебзастосунок не повинен зберігати повні номери карток або CVV-коди на власному сервері - всі дані мають передаватися безпосередньо платіжному шлюзу через захищений токен. Це не лише захищає клієнтів кондитерської, а й знімає з бази практики юридичну відповідальність за зберігання чутливої фінансової інформації.

Під час тестування платіжних сценаріїв розробники повинні використовувати лише «тестові картки» (Sandbox mode). Це виключає ризик

					КвРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		17

випадкових реальних списань та гарантує, що розробка ведеться у безпечному програмному середовищі. У разі виникнення позаштатних ситуацій під час тестування, наприклад, перегріву обладнання при масових запитах, слід негайно діяти згідно з інструкцією протипожежної безпеки.

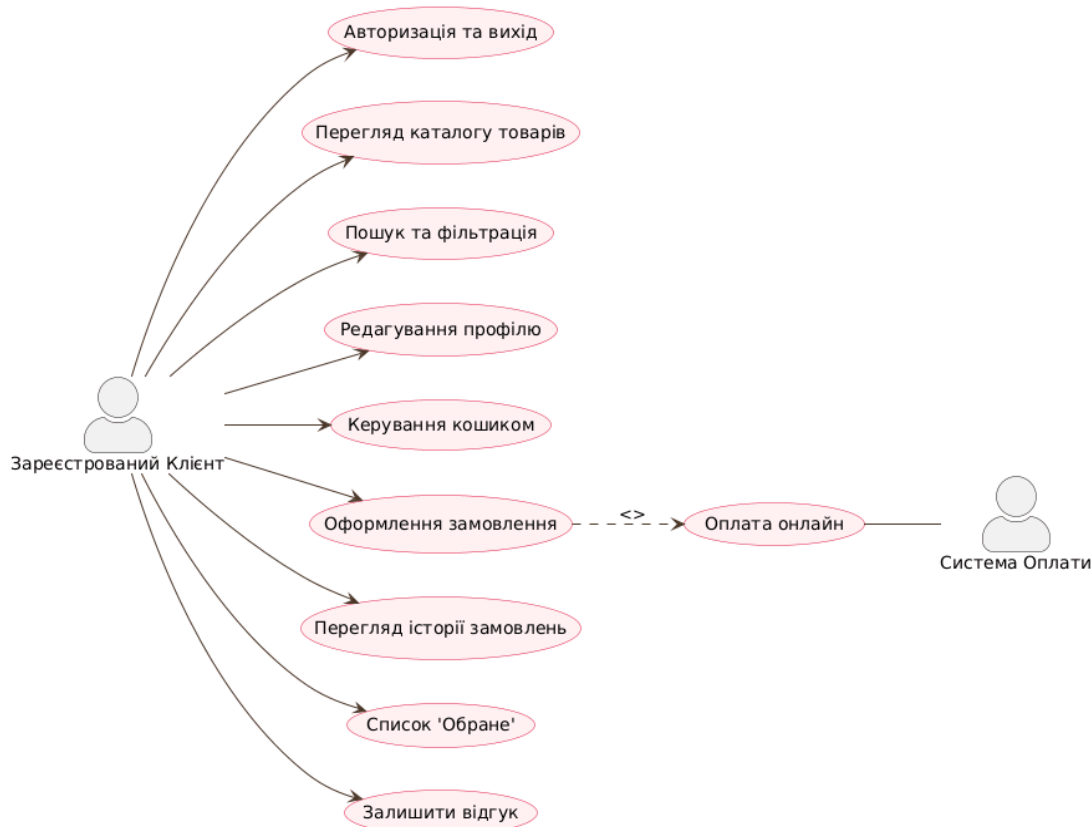


Рисунок 1.1 – Діаграма варіантів використання для зареєстрованого користувача та системи оплати

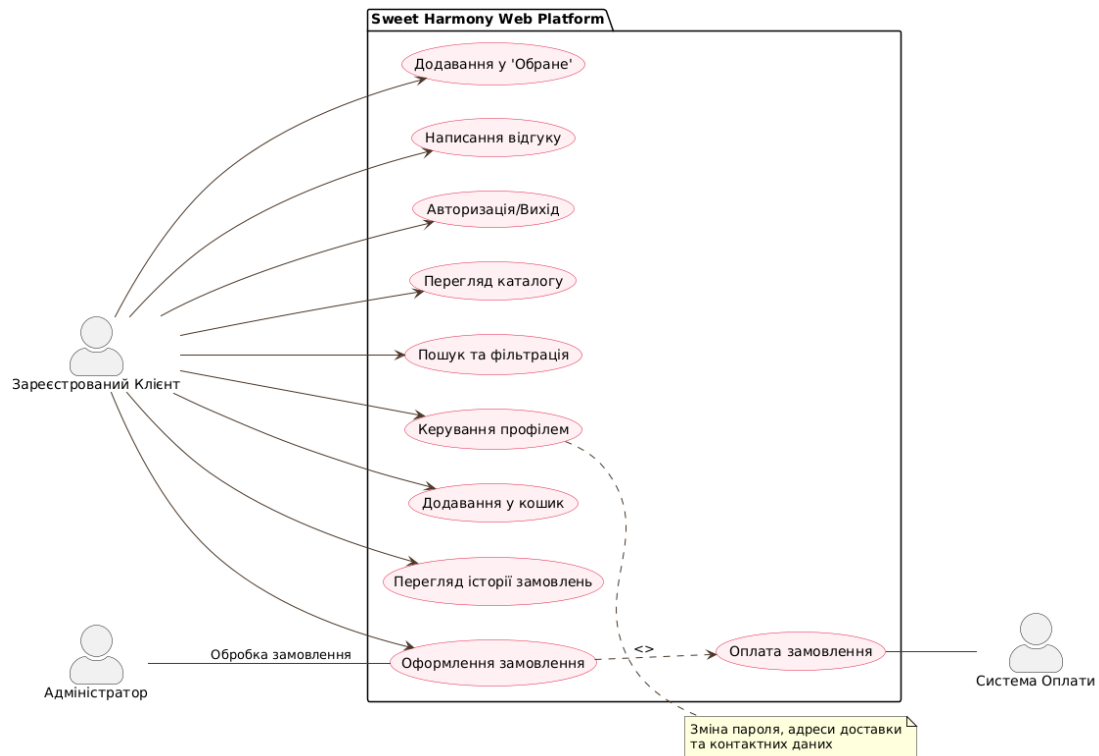


Рисунок 1.2 – Діаграма варіантів використання для зареєстрованого користувача та адміністратора

У даній діаграмі такі дійові особи:

- зареєстрований клієнт, тобто основний користувач, який має доступ до розширеного функціоналу;
- система оплати або зовнішній сервіс, наприклад, LiqPay або Stripe, що взаємодіє з процесом оплати;
- адміністратор, що отримує повідомлення та обробляє замовлення, створені клієнтом.

Основні варіанти використання (Use Cases):

- керування профілем та історія, тобто те, що відрізняє зареєстрованого користувача від гостя, а саме можливість бачити попередні покупки;
- обране (Favorites), що містить функцію збереження товарів на майбутнє;
- оформлення замовлення, що включає процес оплати (<<include>>).

1.4 Висновки. Постановка задачі

Отже, початком роботи можна вважати постановку мети та відповідних завдань для її досягнення.

Метою даної кваліфікаційної роботи є розробка вебзастосунку для продажу кондитерських виробів, що містить функціональні можливості для купівлі солодоців, користуючись мережею Інтернет.

Для того, щоб була досягнута мета потрібно здійснити розв'язок наступних завдань:

- здійснити аналіз ринку реалізації кондитерських виробів онлайн;
- проаналізувати існуючі програмні рішення, що допомагають вести онлайн-бізнес з продажу кондитерських виробів;
- встановити вимоги до розробки програмного забезпечення;
- здійснити розробку технічного завдання;
- проаналізувати та обрати тип архітектури, а також проектування бази даних;
- провести реалізацію вебзастосунку для продажу кондитерських виробів;
- здійснити тестування вебзастосунку для реалізації кондитерських виробів.

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
						20
Зм.	Арк	№ докум.	Підпис	Дата		

2 ПРОЕКТУВАННЯ ВЕБЗАСТОСУНКУ

2.1 Проектування архітектури та структури системи

Для проекту Sweet Harmony проектування базується на принципах високої доступності, безпеки персональних даних та здатності системи витримувати пікові навантаження під час святкових періодів.

Система реалізується на основі мікросервісно-орієнтованого підходу в межах монолітного репозиторію на початковому етапі. Основним архітектурним шаблоном обрано Model-View-ViewModel (MVVM) для клієнтської частини та Clean Architecture для серверної. Це дозволяє ізолювати бізнес-правила від зовнішніх агентів, таких як бази даних або сторонні API (рисунок 2.1).

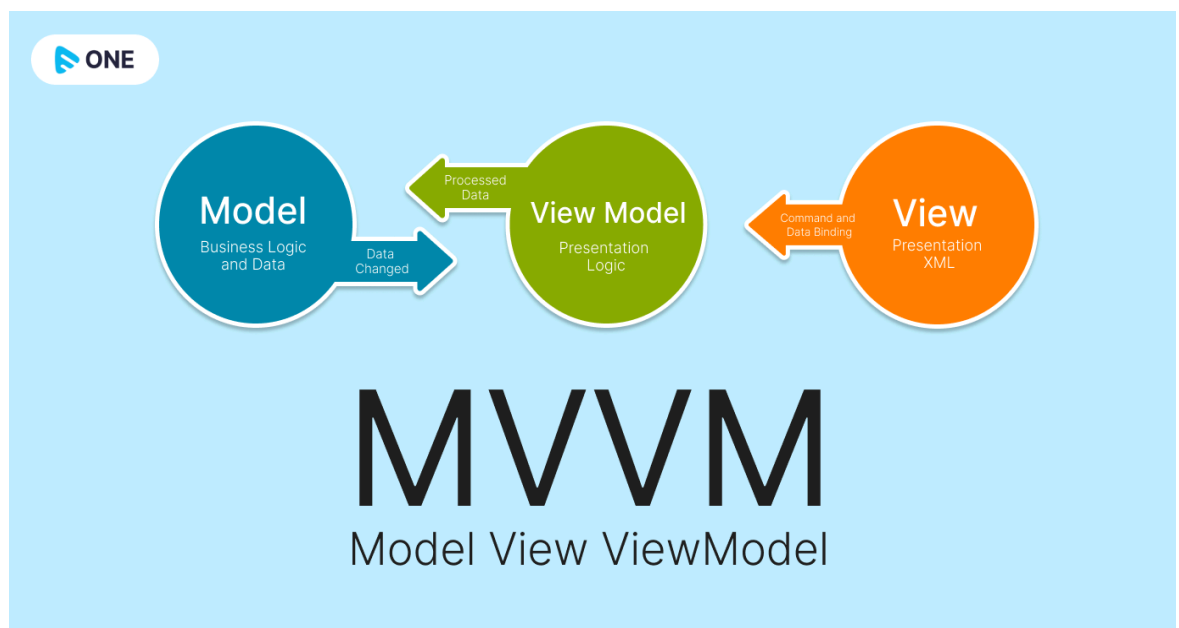


Рисунок 2.1 – Шаблон проектування архітектури MVVM [17]

Рівень представлення та клієнтська логіка полягає у тому, що Frontend будується як Single Page Application (SPA). Весь інтерфейс розділений на атомарні компоненти, що керуються через стан. Для складних замовлень (наприклад, кастомних тортів) впроваджується машина станів (State Machine),

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		21

яка валідує кожен крок користувача перед переходом до оплати. Використання Virtual DOM у поєднанні з оптимізованим рендерингом Tailwind CSS забезпечує миттєвий відгук інтерфейсу навіть при великій кількості графічного контенту.

Backend-частина проектується як набір незалежних сервісів. Сервіс обробки замовлень використовує черги повідомлень для запобігання втрати даних при збоях платіжних шлюзів. Модуль розрахунку логістики інтегрується з картографічними сервісами для динамічного визначення вартості доставки. Безпека забезпечується через багат шарову перевірку токенів та CORS-політики, що обмежують доступ до чутливих методів API.

Структурна організація даних полягає у тому, що проектування бази даних виходить за рамки простих таблиць і включає механізми забезпечення цілісності та швидкості вибірки.

Модель користувачів та безпеки полягає у тому, що сутність User містить не лише базові атрибути, а й розширений профіль з метаданими про вподобання та історію лояльності. Паролі зберігаються виключно у вигляді хешів з використанням алгоритму Argon2. Система передбачає версійність адрес доставки, що дозволяє користувачу зберігати кілька точок отримання без порушення цілісності старих замовлень.

Модель продуктів та складського обліку. Об'єкт Product має складну структуру, що включає масив атрибутів для фільтрації (алергени, рівень цукру, вага). Для підтримки актуальності залишків використовується механізм так званого м'якого бронювання, коли товар додається до кошика на фінальному етапі оформлення, система резервує його на певний час, запобігаючи ситуації оверселінгу.

Логіка замовлень та фінансових транзакцій полягає у тому, що кожне замовлення проходить через чітко визначений життєвий цикл: Pending - Confirmed - In Production - Ready for Delivery - Completed. Зв'язок між замовленням та товарами реалізований через проміжну сутність, яка фіксує

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ док.ум.	Підпис	Дата		22

ціну на момент покупки, що критично для бухгалтерської звітності при зміні цін у каталозі.

Схема компонентної взаємодії допомагає процесу обміну даними між вузлами системи організована наступним чином. Клієнтський додаток ініціює запит до API Gateway, який виконує роль маршрутизатора та контролера навантаження. Запит на авторизацію перенаправляється до модуля Identity Provider, тоді як запити на отримання товарів обслуговуються кешуючим шаром, наприклад, Redis, що значно прискорює відображення каталогу.

Платіжний цикл виділений в окремий ізольований контур. Після ініціації оплати система переводить замовлення в стан очікування та передає керування платіжному шлюзу. Отримання Webhook-підтвердження від банку є тригером для сервісу сповіщень, який автоматично надсилає повідомлення клієнту та на кухонний термінал кондитерської.

Масштабування та відмовостійкість. Система проектується з урахуванням горизонтального масштабування. Статичні ресурси, наприклад, зображення десертів у високій якості розміщуються в Content Delivery Network (CDN), що мінімізує затримки для користувачів з різних географічних локацій. База даних передбачає використання реплік для читання, що дозволяє розвантажити основний вузол під час масового перегляду каталогу.

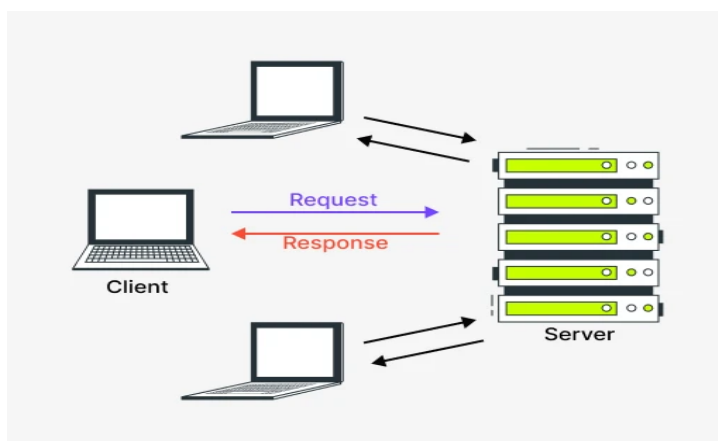


Рисунок 2.2 – Приклад клієнт-серверної архітектури [18]

2.2 Детальне проектування програмного забезпечення

Цей підрозділ описує технічну структуру вебплатформи для продажу кондитерських виробів, базуючись на визначених варіантах використання та вимогах до масштабованості.

До основних модулів системи відносяться:

- модуль каталогу, що відповідає за відображення, фільтрацію за категоріями та пошук десертів;
- модуль кошика, де відбувається тимчасове зберігання обраних товарів у пам'яті клієнта або Redis, розрахунок проміжної суми;
- модуль замовлень, де здійснюється перетворення кошика у фіксований запис у БД, резервування залишків на складі;
- модуль користувача, що здійснює керування особистими даними, перегляд історії замовлень та керування списком Обраного.

Для забезпечення масштабованості, швидкості та сучасного UI обрано наступний стек:

- Frontend - React.js (SPA) + Tailwind CSS (стилізація);
- Backend - Node.js (Express);
- база даних - PostgreSQL (реляційна);
- автентифікація - JWT (JSON Web Tokens);
- платежі через інтеграцію з API (LiqPay).

Система побудована за принципом розподілу відповідальності:

- рівень представлення (Presentation Layer);
- React-компоненти (каталог, кошик, форми авторизації);
- керування станом (React Context);
- рівень бізнес-логіки (Business Logic Layer);
- обробка замовлень;
- розрахунок вартості та знижок;
- валідація даних користувача.

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		24

Рівень доступу до даних (Data Access Layer):

- АРІ-запити до бази даних;
- взаємодія з хмарним сховищем зображень десертів.

2.3 Проектування інтерфейсу користувача вебзастосунку

Для потрібного рівня функціонування розроблюваного програмного забезпечення, необхідно здійснити спочатку проектування, а згодом і реалізацію даного ресурсу. Нижче наведено детальну структуру таблиць реляційної бази даних.

Таблиця 2.1 - Users (Користувачі)

Поле	Тип даних	Опис
user_id	UUID (PK)	Унікальний ідентифікатор користувача
email	VARCHAR(255)	Унікальна адреса (індекс для пошуку)
password_hash	TEXT	Хешований пароль (Argon2)
full_name	VARCHAR(100)	Прізвище та ім'я
phone	VARCHAR(20)	Контактний номер телефону
role	ENUM	Роль: 'client', 'admin'
created_at	TIMESTAMP	Дата та час реєстрації

Таблиця 2.2 - Categories (Категорії)

Поле	Тип даних	Опис
category_id	INT (PK)	Ідентифікатор категорії
title	VARCHAR(100)	Назва (Торти, Макарони тощо)
slug	VARCHAR(100)	URL-ідентифікатор (SEO)

Таблиця 2.3 - Products (Товари)

Поле	Тип даних	Опис
product_id	UUID (PK)	Унікальний код виробу
category_id	INT (FK)	Зв'язок з таблицею Categories
name	VARCHAR(150)	Найменування десерту
description	TEXT	Опис інгредієнтів та особливостей
price	DECIMAL(10,2)	Поточна вартість виробу
stock_quantity	INT	Залишок на складі
image_url	VARCHAR(255)	Посилання на зображення в CDN
is_active	BOOLEAN	Статус доступності для продажу

Таблиця 2.4 - Orders (Замовлення)

Поле	Тип даних	Опис
order_id	UUID (PK)	Номер замовлення
user_id	UUID (FK)	Посилання на клієнта (може бути NULL)
status	ENUM	Статус: 'Pending', 'Paid', 'Delivered', etc.
total_price	DECIMAL(12,2)	Підсумкова сума замовлення
shipping_addr	TEXT	Повна адреса доставки
payment_id	VARCHAR(100)	Ідентифікатор транзакції платіжної системи
created_at	TIMESTAMP	Дата і час створення замовлення

Таблиця 2.5 - Order_Items (Склад замовлення)

Поле	Тип даних	Опис
item_id	BIGINT (PK)	Ідентифікатор рядка

Продовження таблиці 2.5:

order_id	UUID (FK)	Зв'язок з таблицею Orders
product_id	UUID (FK)	Зв'язок з таблицею Products
quantity	INT	Кількість одиниць товару
unit_price	DECIMAL(10,2)	Фіксована ціна на момент купівлі

Таблиця 2.6 - Reviews (Відгуки)

Поле	Тип даних	Опис
review_id	UUID (PK)	Ідентифікатор відгуку
product_id	UUID (FK)	Товар, який оцінюють
user_id	UUID (FK)	Автор відгуку
rating	INT (1-5)	Числова оцінка
comment	TEXT	Текстовий зміст відгуку

На рисунку 2.3 запропоновано схему взаємодії компонентів, що змодельовано у вигляді ER-діаграми.

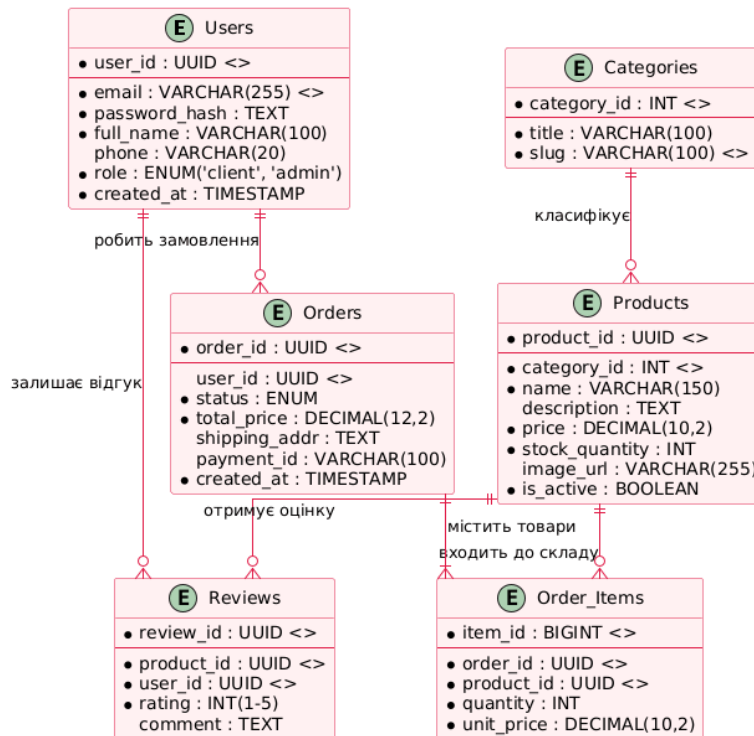


Рисунок 2.3 – ER-діаграма

Вказана діаграма відображає основні типи зв'язків:

1. Один до багатьох (1:N), наприклад, один користувач може мати багато замовлень, або одна категорія містити багато товарів.

2. Багато до багатьох (M:N), що реалізовано через проміжну таблицю ORDER_ITEMS, що пов'язує замовлення та продукти.

2.4 Аналіз та вибір технологій і методів реалізації програмного забезпечення.

У даному підрозділі описується набір інструментів, що використовуються для розробки вебресурсу для продажу кондитерських виробів. Вибір технологій базується на необхідності красивої візуальної презентації товарів, а саме солодоців, високих вимогах до SEO, швидкості роботи та безпеки онлайн-платежів.

Клієнтська частина орієнтована на створення привабливого та зручного інтерфейсу з акцентом на швидке завантаження візуального контенту. Основним фреймворком виступає Next.js на базі React. Це ідеальний вибір для електронної комерції завдяки технології Server-Side Rendering (SSR), яка забезпечує відмінну SEO-оптимізацію та дозволяє пошуковим системам легко індексувати сторінки з тортами. Крім того, Next.js має вбудовану оптимізацію зображень. Увесь код пишеться на TypeScript, що завдяки суворій типізації допомагає уникати помилок, особливо при роботі з цінами та логікою кошика. Для швидкого створення адаптивного та унікального дизайну застосовується Tailwind CSS. Управління глобальним станом покупця, насамперед його кошиком, здійснюється за допомогою Zustand або Redux Toolkit, тоді як за асинхронні дані, кешування каталогу товарів та швидку синхронізацію з сервером відповідає React Query.

Серверна частина відповідає за бізнес-логіку, обробку замовлень, розрахунок вартості доставки та безпеку транзакцій. В якості швидкого та

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		28

масштабованого середовища для створення API використовується Node.js разом із фреймворками NestJS. Критично важливою складовою є безпечна оплата, тому реалізована інтеграція з платіжними шлюзами, такими як LiqPay, WayForPay, Fondy або Stripe, що підтримують оплату картками та Apple Pay / Google Pay. Авторизація клієнтів для доступу до особистого кабінету та історії замовлень відбувається на базі безпечних токенів JWT. Також налаштована пряма інтеграція з API поштових сервісів, наприклад, Нової Пошти, для точного розрахунку вартості доставки під час оформлення замовлення.

Меню кондитерської потребує регулярного оновлення: додавання сезонних десертів, зміна цін та описів. Для цього впроваджено Headless CMS, таку як Strapi або Sanity. Ці системи надають зручну адміністративну панель, через яку менеджери та кондитери можуть легко управляти каталогом товарів, характеристиками (склад, алергени) та фотографіями без жодного залучення розробників.

Зберігання даних організовано з урахуванням швидкодії та надійності. Основною реляційною базою даних для фіксації замовлень, профілів клієнтів та аналітики виступає PostgreSQL, а взаємодія з нею ведеться через зручний інструмент Prisma ORM. Для того, щоб навігація сайтом була миттєвою, каталог десертів додатково кешується в оперативній пам'яті за допомогою Redis. Оскільки кондитерські вироби потребують якісної візуальної подачі, усі важкі медіафайли та фотографії зберігаються у спеціалізованому хмарному сховищі AWS S3 або Cloudinary, що гарантує їхню швидку віддачу користувачам.

Процеси розгортання та підтримки інфраструктури налаштовані для максимальної стабільності. Фронтенд розміщується на хмарній платформі Vercel, яка завдяки глобальному CDN забезпечує миттєве завантаження сайту з будь-якого регіону. Серверна частина та бази даних розгортаються ізольовано за допомогою контейнерів Docker на потужностях DigitalOcean або AWS. Для безперервної інтеграції та доставки коду використовуються

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		29

пайплайни GitHub Actions, які автоматизують тестування та оновлення проєкту.

Стабільність інтернет-магазину підтримується багаторівневим тестуванням. Критична бізнес-логіка, зокрема правильність розрахунку загальної суми в кошику та застосування промокодів, перевіряється модульними тестами за допомогою Jest. Головні користувацькі шляхи - від відкриття каталогу і додавання товару в кошик до успішного завершення покупки - контролюються автоматизованими E2E-тестами в середовищі Cypress.

У даному підрозділі міститься опис концептуального та структурного проектування користувацького інтерфейсу вебзастосунку кондитерської «Sweet Harmony». Головною метою інтерфейсу є створення емоційного зв'язку з клієнтом через яскравий та цікавий візуальний ряд, забезпечивши при цьому максимально простий та інтуїтивний шлях до покупки. Для цього здійснюється розробка візуальної концепції та дизайн-системи, вибір глобальних компонентів інтерфейсу, проектування ключових екранів, мікроінтеракції та адаптивність.

1. Візуальна концепція та дизайн-система.

Візуальний стиль платформи базується на відчутті преміальності, затишку та натуральності, де інтерфейс виконує роль витонченої рамки, яка не відвертає увагу від фотографій десертів.

Колірна палітра побудована навколо теплого кремowego відтінку основного фону, який асоціюється з ваніллю та вершками, знижуючи навантаження на очі порівняно з агресивним білим кольором. Для привернення уваги та стимулювання апетиту використовується акцентний ягідно-рожевий колір. Він застосовується виключно для головних цільових кнопок, іконок кошика та бейджів зі знижками. Текстова складова оформлена

					КвРІПЗ.2201112.01.17.ПЗ	Арк.
						30
Зм.	Арк	№ докум.	Підпис	Дата		

у глибокому коричневому відтінку темного шоколаду для заголовків та м'якому сірому для другорядних описів, що забезпечує високий контраст і читабельність без зайвої суворості чорного кольору.

Типографіка системи спирається на контраст двох сімейств шрифтів. Заголовки оформлені елегантним шрифтом із зарубками, таким як Playfair Display, що підкреслює статусність та ремісничий характер виробів. Основний текст, описи товарів та інтерфейсні елементи використовують чистий гротеск, наприклад, Inter, для ідеальної читабельності навіть у найдрібніших деталях.

2. Глобальні компоненти інтерфейсу.

Верхня навігаційна панель є ключовим елементом орієнтації користувача. Вона закріплена у верхній частині екрана та має властивість легкого розмиття фону при скролінгу. Структурно вона поділена на три зони: зліва розташований логотип, що слугує точкою повернення на головну сторінку; по центру розміщене основне навігаційне меню з посиланнями на каталог, інформацію про компанію та умови доставки; справа знаходиться панель користувача з іконкою профілю та динамічним індикатором кошика, який відображає кількість обраних товарів.

Замість традиційного переходу на окрему сторінку кошика, система використовує висувну бічну панель. При натисканні на відповідну іконку вона плавно з'являється з правого краю екрана. Це архітектурне рішення дозволяє користувачеві переглядати обрані товари, змінювати їх кількість та бачити загальну вартість замовлення, не втрачаючи при цьому контексту каталогу, що значно покращує загальний досвід покупок.

3. Проектування ключових екранів.

Головна сторінка слугує вітриною закладу та виконує роль потужного емоційного гачка. Вона відкривається великим фоновим зображенням або відео процесу створення десертів, поверх якого розміщено головний мотиваційний заголовок та кнопку переходу до покупок. Далі сторінка плавно перетікає у блок з найпопулярнішими десертами та інформаційну секцію, яка

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		31

через лаконічну інфографіку розповідає про переваги бренду: натуральні інгредієнти, ручну роботу та швидку доставку.

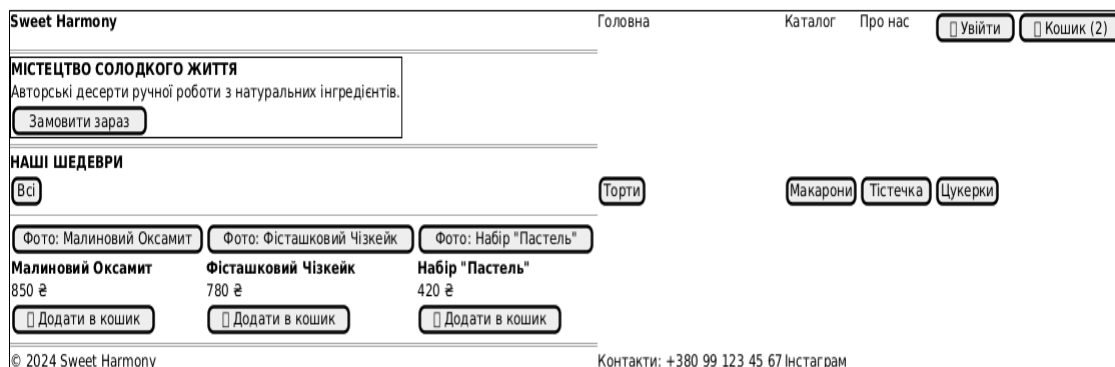


Рисунок 2.4 – Макет головного екрану

Сторінка каталогу спроектована для максимально зручного вибору. У верхній частині розташована горизонтальна панель фільтрів у вигляді кнопок-тегів для швидкого перемикання між категоріями. Під нею розміщена сітка карток товарів. Кожна картка містить велике фото з ефектом легкого збільшення при наведенні курсору, назву, ціну, короткий опис та кнопки для швидкого додавання в кошик або збереження до списку обраного.

Макет Висувна панель кошика (рисунок 2.5) демонструє бічну панель кошика, яка дозволяє переглядати обрані товари, змінювати їхню кількість та бачити загальну суму без переходу на іншу сторінку.



Рисунок 2.5 – Макет висувного кошика

Детальний опис товару відкривається на окремій сторінці, яка візуально поділена на дві частини. Ліва відведена під розширену галерею з основним фото та мініатюрами інших ракурсів десерту. Права частина містить усю необхідну інформацію: від складу та виділених алергенів до інтерактивних селекторів розміру, лічильника кількості та великої кнопки покупки. У нижній частині сторінки розміщено блок соціального доказу — відгуки попередніх клієнтів та середній рейтинг товару.

Екран оформлення замовлення концептуально ізольований від зайвої навігації, щоб сфокусувати увагу клієнта та мінімізувати відсоток покинутих кошиків. Процес розбито на інтуїтивні кроки: введення контактних даних, налаштування параметрів доставки з вибором часу через зручний віджет календаря, та вибір методу оплати. У правій частині екрана постійно закріплений блок із підсумком замовлення, який забезпечує прозорість розрахунків. Цей мокап відображає ізольований процес покупки. Екран розділений на дві колонки: ліворуч - кроки для заповнення інформації, наприклад, контакти, доставка, оплата, праворуч - фіксований підсумок замовлення (рисунок 2.6).

Sweet Harmony		Оформлення безпечного замовлення	
1. Контактні дані		ВАШЕ ЗАМОВЛЕННЯ	
Ім'я:	<input type="text" value="Олександр"/>	Малиновий Оксамит (1)	850 ₴
Телефон:	<input type="text" value="+380 99 123 45 67"/>	Набір "Пастель" (1)	420 ₴
Email:	<input type="text" value="alex@example.com"/>	Доставка	0 ₴
2. Доставка		До сплати:	1270 ₴
Спосіб:	<input type="radio"/> Кур'єром <input type="radio"/> Самовивіз	<input type="button" value="Підтвердити замовлення"/>	
Адреса:	<input type="text" value="вул. Шоколадна, 12"/>		
Час:	<input type="text" value="Сьогодні 15:00 - 17:00"/>		
3. Оплата			
Метод:	<input type="radio"/> Карткою онлайн <input type="radio"/> Готівкою при отриманні		

Рисунок 2.6 – Екран оформлення замовлення

Особистий кабінет, доступний після авторизації, пропонує зручне бічне меню. Через нього користувач отримує швидкий доступ до історії своїх замовлень із можливістю повторити попередню покупку в один клік,

керування особистими налаштуваннями, адресами доставки та перегляду відкладених товарів у списку обраного.

4. Мікроінтеракції та адаптивність.

Якісний інтерфейс повинен вести безперервний діалог з користувачем через систему зворотного зв'язку. Процес додавання в кошик супроводжується миттєвою візуальною реакцією: кнопка змінює свій стан, іконка в навігаційній панелі анімовано реагує на новий товар, а внизу екрана з'являється коротке підтвердуюче сповіщення. Під час завантаження даних застосовуються сірі блоки-заглушки (скелетони), що імітують контент та створюють ілюзію швидшої роботи системи. Валідація форм відбувається в режимі реального часу, миттєво підсвічуючи помилки введення без необхідності перезавантаження сторінки.

Оскільки лєвова частка замовлень здійснюється з мобільних пристроїв, дизайн розробляється за принципом Mobile-First. Навігація трансформується у компактне приховане меню, сітка товарів адаптується для зручного скролінгу великим пальцем, а найважливіші кнопки - додавання в кошик та оформлення замовлення - займають всю ширину екрана та фіксуються в нижній, найбільш доступній зоні дисплея.

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		34

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

3.1 Реалізація бази даних

Для якісного розроблення програмного застосунку необхідно здійснити реалізацію бази даних. Послідовність реалізації бази даних «Sweet Harmony» заснована на реалізації реляційної бази даних на прикладі наданого SQL-скрипта, що вимагає чіткої послідовності. Головне правило полягає у тому, що не можна створити об'єкт, наприклад, таблицю, який посилається на інший об'єкт, поки цей інший об'єкт не існує.

Детальний алгоритм реалізації складається із декількох кроків, а саме підготовки середовища та розширень, створення кастомних типів (ENUM), створення незалежних (базових) таблиць, створення таблиць 1-го рівня залежності, створення таблиць 2-го рівня залежності, тобто зв'язуючих таблиць, налаштування індексів, наповнення початковими даними (Seeding).

Підготовка середовища та розширень вимагає необхідної підготовки інфраструктури бази даних. Для безпосереднього створення самої бази даних на сервері PostgreSQL створюється порожня база, наприклад, `CREATE DATABASE sweet_harmony`.

Активація розширень: Виконується команда `CREATE EXTENSION IF NOT EXISTS «uuid-ossr»`. Це робиться в першу чергу, оскільки більшість наших таблиць використовують UUID як первинний ключ (PRIMARY KEY), і базі потрібен алгоритм для їх генерації (`uuid_generate_v4()`).

Крок 2: Створення кастомних типів (ENUM).

Наступним кроком є створення перелічень (ENUM), оскільки вони будуть використовуватися як типи даних для колонок у майбутніх таблицях.

Створюється `user_role` (клієнт, адмін).

Створюється `order_status` (статуси від очікування до доставки).

Якщо спробувати створити таблицю `users` до створення типу `user_role`, база даних видасть помилку.

Крок 3: Створення незалежних (базових) таблиць.

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		35

Тепер створюються таблиці верхнього рівня, які не мають зовнішніх ключів і не залежать від інших таблиць.

users (Користувачі) створюється першою, адже до неї будуть прив'язуватись замовлення та відгуки.

categories (Категорії) створюється зараз, оскільки без категорій ми не зможемо додати жодного товару.

Крок 4: Створення таблиць 1-го рівня залежності.

Далі йдуть таблиці, які посилаються на базові таблиці (Крок 3).

products (Товари) вимагає існування categories, оскільки поле category_id є зовнішнім ключем.

orders (Замовлення) вимагає існування users, щоб прив'язати замовлення до конкретного клієнта (user_id).

Крок 5: Створення таблиць 2-го рівня залежності (Зв'язуючі таблиці).

На цьому етапі створюються найскладніші таблиці, які пов'язують між собою кілька інших сутностей (відношення «багато-до-багатьох»).

order_items (Склад замовлення) може бути створена лише тоді, коли існують і orders (замовлення створено), і products (товар існує).

reviews (Відгуки) вимагає існування і users (хто залишає відгук), і products (на що залишають).

Крок 6: Налаштування індексів.

Після того, як структуру створено, додаються індекси (CREATE INDEX). Вони створюються для тих колонок, за якими найчастіше буде відбуватися пошук (наприклад, фільтрація товарів за category_id або пошук активних товарів is_active). Індокси застосовуються після створення таблиць, щоб база даних одразу правильно побудувала внутрішні дерева пошуку.

Крок 7: Наповнення початковими даними (Seeding).

Останній крок реалізації (ініціалізації) - це вставка базових або тестових даних (INSERT INTO). Тут також діє правило залежностей, коли спочатку заповнюється categories, тобто додається підрозділ «Торти», «Макарони».

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		36

Тільки після цього заповнюємо *products*, використовуючи *category_id* щойно створених категорій (за допомогою підзапиту `SELECT category_id FROM categories WHERE slug = 'cakes'`).

Якщо спробувати виконати SQL-скрипт у випадковому порядку, наприклад, спочатку створити *order_items*, а потім *orders*, PostgreSQL заблокує операцію з помилкою типу `relation «orders» does not exist`. Запропонована структура гарантує безконфліктне розгортання БД з першого разу.

Скрипт ініціалізації бази даних для кондитерської «Sweet Harmony» (PostgreSQL) показано нижче.

Підключення розширення для автоматичної генерації UUID
`CREATE EXTENSION IF NOT EXISTS «uuid-oss»;`

1. Створення власних типів даних (ENUM)

```
CREATE TYPE user_role AS ENUM ('client', 'admin');
```

```
CREATE TYPE order_status AS ENUM ('Pending', 'Paid', 'Processing',  
'Shipped', 'Delivered', 'Cancelled');
```

2. Створення таблиць

Таблиця користувачів (клієнти та адміністратори)

```
CREATE TABLE users (  
    user_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
    email VARCHAR(255) UNIQUE NOT NULL,  
    password_hash TEXT NOT NULL,  
    full_name VARCHAR(100) NOT NULL,  
    phone VARCHAR(20),  
    role user_role DEFAULT 'client' NOT NULL,  
    created_at TIMESTAMP WITH TIME ZONE DEFAULT  
CURRENT_TIMESTAMP
```

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		37

);

Таблиця категорій десертів

```
CREATE TABLE categories (  
    category_id SERIAL PRIMARY KEY,  
    title VARCHAR(100) NOT NULL,  
    slug VARCHAR(100) UNIQUE NOT NULL  
);
```

Таблиця товарів (асортимент)

```
CREATE TABLE products (  
    product_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
    category_id INT NOT NULL,  
    name VARCHAR(150) NOT NULL,  
    description TEXT,  
    price DECIMAL(10, 2) NOT NULL CHECK (price >= 0),  
    stock_quantity INT NOT NULL DEFAULT 0 CHECK (stock_quantity >=  
0),  
    image_url VARCHAR(255),  
    is_active BOOLEAN DEFAULT TRUE,
```

Зв'язок з категоріями (якщо категорію видаляють, товари теж видаляються)

```
CONSTRAINT fk_category  
    FOREIGN KEY (category_id)  
    REFERENCES categories(category_id)  
    ON DELETE CASCADE  
);
```

Таблиця замовлень (дані з екрана Checkout)

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		38

```

CREATE TABLE orders (
    order_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    user_id UUID, -- Може бути NULL для гостей
    status order_status DEFAULT 'Pending' NOT NULL,
    total_price DECIMAL(12, 2) NOT NULL CHECK (total_price >= 0),
    shipping_addr TEXT NOT NULL,
    payment_id VARCHAR(100), -- ID транзакції від платіжного шлюзу
    created_at    TIMESTAMP    WITH    TIME    ZONE    DEFAULT
CURRENT_TIMESTAMP,

```

Якщо користувач видаляє акаунт, замовлення залишається в системі (SET NULL)

```

CONSTRAINT fk_user
    FOREIGN KEY (user_id)
    REFERENCES users(user_id)
    ON DELETE SET NULL
);

```

Таблиця елементів замовлення (зв'язок багато-до-багатьох між замовленнями та товарами)

```

CREATE TABLE order_items (
    item_id BIGSERIAL PRIMARY KEY,
    order_id UUID NOT NULL,
    product_id UUID NOT NULL,
    quantity INT NOT NULL CHECK (quantity > 0),
    unit_price DECIMAL(10, 2) NOT NULL CHECK (unit_price >= 0), --
Фіксується ціна на момент покупки

```

Видалення замовлення каскадно видаляє його елементи

```

CONSTRAINT fk_order

```

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		39

```
FOREIGN KEY (order_id)
REFERENCES orders(order_id)
ON DELETE CASCADE,
```

Не можна видалити товар, якщо він присутній в оформлених замовленнях (RESTRICT)

```
CONSTRAINT fk_product
FOREIGN KEY (product_id)
REFERENCES products(product_id)
ON DELETE RESTRICT
);
```

Таблиця відгуків

```
CREATE TABLE reviews (
review_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
product_id UUID NOT NULL,
user_id UUID NOT NULL,
rating INT NOT NULL CHECK (rating >= 1 AND rating <= 5),
comment TEXT,
created_at TIMESTAMP WITH TIME ZONE DEFAULT
CURRENT_TIMESTAMP,
```

Каскадне видалення відгуків при видаленні товару або користувача

```
CONSTRAINT fk_review_product
FOREIGN KEY (product_id)
REFERENCES products(product_id)
ON DELETE CASCADE,
CONSTRAINT fk_review_user
FOREIGN KEY (user_id)
REFERENCES users(user_id)
```

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		40

ON DELETE CASCADE

);

3. Створення індексів для прискорення запитів

Індекси для швидкого пошуку та фільтрації в каталозі

```
CREATE INDEX idx_products_category ON products(category_id);
```

```
CREATE INDEX idx_products_is_active ON products(is_active);
```

Індекси для особистого кабінету та адмін-панелі

```
CREATE INDEX idx_orders_user ON orders(user_id);
```

```
CREATE INDEX idx_orders_status ON orders(status);
```

```
CREATE INDEX idx_order_items_order ON order_items(order_id);
```

4. Тестові дані (Seed Data)

```
INSERT INTO categories (title, slug) VALUES
```

```
('Торти', 'cakes'),
```

```
('Макарони', 'macarons'),
```

```
('Тістечка', 'pastries');
```

```
INSERT INTO products (category_id, name, description, price, stock_quantity, is_active) VALUES
```

```
((SELECT category_id FROM categories WHERE slug = 'cakes'), 'Малиновий Оксамит', 'Ніжний бісквіт з малиновим конфі', 850.00, 10, true),
```

```
((SELECT category_id FROM categories WHERE slug = 'macarons'), 'Набір «Пастель», '12 вишуканих макаронів', 420.00, 25, true);
```

Реалізація реляційної бази даних вимагає суворого дотримання черговості виконання SQL-команд. Фундаментальне правило полягає в тому, що неможливо створити таблицю або зв'язок, який посилається на ще не

					КвРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		41

існуючий об'єкт бази даних. Тому процес розгортання відбувається послідовно, від найпростіших незалежних елементів до складних зв'язків.

Розгортання починається з підготовки самого середовища. Насамперед активується розширення для генерації унікальних ідентифікаторів, оскільки наша архітектура використовує формат UUID для первинних ключів більшості таблиць. Одразу після цього система ініціалізує користувацькі типи даних, такі як перелічення для ролей користувачів та статусів замовлень. Створення цих типів на самому початку є критично важливим, адже вони будуть використовуватися як складові частини колонок у майбутніх таблицях.

Наступним етапом є формування так званих «незалежних» або базових таблиць. Це сутності користувачів та категорій десертів. Вони створюються найпершими, оскільки не містять жодних зовнішніх ключів і абсолютно не залежать від існування інших елементів системи. Вони виконують роль надійного фундаменту для подальшої побудови архітектури.

Маючи готову базу, ми переходимо до створення таблиць першого рівня залежності. На цьому кроці вводиться таблиця товарів, яка фізично не може існувати без попередньо створеної таблиці категорій, оскільки має на неї пряме посилання. Паралельно створюється таблиця замовлень, яка за аналогічним принципом жорстко прив'язується до вже існуючої таблиці користувачів.

Після закріплення цих сутностей настає черга складних перехресних зв'язків. Система формує таблиці другого рівня залежності, які поєднують між собою відразу кілька попередньо створених об'єктів. Зокрема, ініціалізується таблиця деталей замовлення, яка одночасно спирається і на загальну таблицю замовлень, і на таблицю товарів. За таким же правилом створюється структура для зберігання відгуків, яка об'єднує користувачів із товарами, що вони оцінюють.

Коли весь каркас таблиць та зовнішніх ключів успішно зведено, розпочинається етап оптимізації продуктивності. База даних генерує індекси для тих колонок, за якими передбачається найінтенсивніший пошук,

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		42

наприклад, ідентифікатори категорій або статуси замовлень. Побудова індексів саме на цьому етапі дозволяє базі даних одразу правильно структурувати внутрішні дерева пошуку інформації.

Фінальним акордом ініціалізації є наповнення бази стартовими даними. Цей процес також суворо підпорядковується правилу залежностей: система спочатку вставляє базові категорії, і лише після їх успішного збереження додає тестові товари, динамічно посилаючись на щойно згенеровані ідентифікатори цих категорій. Після завершення цього кроку розгортання вважається успішним, а база даних - готовою до роботи.

3.2 Реалізація модулів системи

Даний підрозділ описує внутрішню логіку роботи та етапи технічної реалізації основних програмних модулів системи. Кожен модуль відповідає за окрему бізнес-функцію і тісно взаємодіє зі спроектованою раніше реляційною базою даних PostgreSQL.

Модуль автентифікації та управління користувачами. Реалізація цього модуля базується на безпечній передачі та зберіганні конфіденційних даних клієнтів. Коли новий користувач заповнює форму реєстрації на клієнтській частині, React-додаток відправляє ці дані на сервер через захищене з'єднання. Бекенд ніколи не зберігає паролі у відкритому вигляді. Замість цього він використовує криптографічний алгоритм, наприклад bcrypt або Argon2, для створення необоротного хешу, який і записується у таблицю користувачів бази даних.

Приклад бекенд-контролера для реєстрації:

```
const bcrypt = require('bcrypt');  
const jwt = require('jsonwebtoken');  
  
async function registerUser(req, res) {
```

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
						43
Зм.	Арк	№ докум.	Підпис	Дата		

```

const { email, password, fullName } = req.body;

// Хешування пароля
const saltRounds = 10;
const passwordHash = await bcrypt.hash(password, saltRounds);

try {
  // Збереження користувача в БД
  const result = await db.query(
    `INSERT INTO users (email, password_hash, full_name)
    VALUES ($1, $2, $3) RETURNING user_id, role`,
    [email, passwordHash, fullName]
  );

  const user = result.rows[0];

  // Генерація JWT токена
  const token = jwt.sign(
    { userId: user.user_id, role: user.role },
    process.env.JWT_SECRET,
    { expiresIn: '24h' }
  );

  res.status(201).json({ token, message: "Реєстрація успішна" });
} catch (error) {
  res.status(500).json({ error: "Помилка при реєстрації" });
}
}

```

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		44

Процес авторизації побудований на використанні JSON Web Tokens (JWT). Після успішної перевірки електронної пошти та хешу пароля сервер генерує унікальний токен доступу. Цей токен повертається клієнту і зберігається в безпечних HTTP-only файлах cookie або локальному сховищі браузера. При кожному наступному запиті до захищених маршрутів, наприклад, при спробі змінити номер телефону або переглянути історію замовлень, фронтенд автоматично додає цей токен до заголовків запиту. Сервер перевіряє валідність токена і лише після цього виконує операцію оновлення даних у базі.

Модуль каталогу та управління товарами. Модуль каталогу є найбільш навантаженою частиною системи, оскільки саме з ним взаємодіють усі відвідувачі сайту. Для забезпечення швидкого завантаження карток десертів, клієнтський додаток не завантажує всю базу товарів одразу. Замість цього реалізовано механізм пагінації або нескінченного скролінгу. Фронтенд надсилає запит на отримання обмеженої кількості товарів, а бекенд формує відповідний SQL-запит до таблиці продуктів, використовуючи ліміти та зміщення.

Приклад SQL-запиту для отримання товарів з пагінацією та фільтрацією:

SELECT

p.product_id,

p.name,

p.price,

p.image_url,

c.title as category_name

FROM products p

JOIN categories c ON p.category_id = c.category_id

WHERE

p.is_active = true

AND (c.slug = \$1 OR \$1 IS NULL) -- Фільтр за категорією, якщо

передано

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		45

ORDER BY p.created_at DESC

LIMIT \$2 OFFSET \$3; -- Пагінація (наприклад, LIMIT 12 OFFSET 0)

Функція фільтрації за категоріями працює шляхом передачі ідентифікатора категорії у параметрах URL-адреси. Отримавши такий запит, сервер робить вибірку з бази даних, приєднуючи таблицю категорій та відсіюючи товари, які не відповідають критерію або мають статус неактивних. Самі зображення десертів не зберігаються безпосередньо в базі даних. У таблиці міститься лише текстове посилання на файл, який фізично знаходиться на серверах швидкої доставки контенту (CDN). Це дозволяє суттєво зменшити навантаження на основний сервер та прискорити рендеринг сторінки у браузері клієнта.

Модуль кошика та оформлення замовлень. Реалізація кошика розділена на два етапи для оптимізації продуктивності. На етапі вибору товарів кошик існує виключно в оперативній пам'яті клієнтського браузера. Для цього використовується контекст стану React (React Context), який дозволяє миттєво перераховувати проміжну суму та оновлювати іконку кошика без жодного звернення до сервера бази даних.

Приклад транзакції оформлення замовлення:

```
async function createOrder(userId, cartItems, shippingAddr, totalPrice) {
  const client = await pool.connect();
  try {
    await client.query('BEGIN'); // Початок транзакції

    // Створення запису замовлення
    const orderRes = await client.query(
      `INSERT INTO orders (user_id, total_price, shipping_addr, status)
      VALUES ($1, $2, $3, 'Pending') RETURNING order_id`,
      [userId, totalPrice, shippingAddr]
    );
  }
```

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		46

```

const orderId = orderRes.rows[0].order_id;

// Додавання товарів та списання залишків
for (let item of cartItems) {
    // Запису деталей за фіксованою ціною
    await client.query(
        `INSERT INTO order_items (order_id, product_id, quantity,
unit_price)
        VALUES ($1, $2, $3, $4)`,
        [orderId, item.productId, item.quantity, item.price]
    );

    // Зменшуємо залишок на складі
    await client.query(
        `UPDATE products SET stock_quantity = stock_quantity - $1
        WHERE product_id = $2`,
        [item.quantity, item.productId]
    );
}

await client.query('COMMIT'); // Успішне завершення транзакції
return orderId;
} catch (e) {
    await client.query('ROLLBACK'); // Відкат змін у разі помилки
    throw e;
} finally {
    client.release();
}
}

```

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		47

Критична фаза починається у момент натискання кнопки підтвердження замовлення на екрані оформлення. Фронтенд формує єдиний пакет даних із вмістом кошика та адресою доставки і відправляє його на сервер. Бекенд відкриває транзакцію в базі даних PostgreSQL, що гарантує цілісність операції. Спочатку сервер перевіряє наявність потрібної кількості товарів на складі, звертаючись до таблиці продуктів. Якщо перевірка успішна, він створює новий запис у таблиці замовлень зі статусом очікування. Одразу після цього система переносить товари з клієнтського кошика у таблицю деталей замовлення бази даних. Дуже важливим аспектом є те, що сервер бере ціну не з даних клієнта, а з власної бази, і жорстко фіксує її в деталях замовлення, захищаючи чек від майбутніх змін вартості. Наприкінці транзакції система віднімає куплену кількість одиниць від загального залишку на складі.

Модуль платіжних інтеграцій. Оскільки система не обробляє дані банківських карток напряму заради безпеки, модуль платежів працює як посередник між нашим додатком та стороннім платіжним шлюзом. Після успішного створення замовлення в базі даних, наш бекенд формує спеціальний криптографічний підпис і відправляє запит до API платіжної системи, наприклад LiqPay або Stripe. У відповідь система отримує унікальне посилання на захищену сторінку оплати, куди фронтенд автоматично перенаправляє клієнта.

Приклад обробника Webhook від платіжної системи:

```
app.post('/api/webhook/payment', express.raw({type: 'application/json'}),
async (req, res) => {
  const signature = req.headers['stripe-signature'];
  let event;

  try {
    // Перевірка криптографічного підпису для підтвердження, що
    запит від банку
```

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
						48
Зм.	Арк	№ докум.	Підпис	Дата		

```

        event = stripe.webhooks.constructEvent(req.body, signature,
process.env.STRIPE_WEBHOOK_SECRET);
    } catch (err) {
        return res.status(400).send(`Webhook Error: ${err.message}`);
    }

    // Якщо оплата пройшла успішно
    if (event.type === 'checkout.session.completed') {
        const session = event.data.object;
        const orderId = session.client_reference_id;
        const paymentId = session.payment_intent;

        // Оновлення статусу замовлення в БД
        await db.query(
            `UPDATE orders SET status = 'Paid', payment_id = $1 WHERE
order_id = $2`,
            [paymentId, orderId]
        );

        // Тут можна додати логіку відправки email клієнту
    }

    res.json({received: true});
});

```

Поки клієнт вводить дані картки на стороні банку, наше замовлення зберігає статус очікування. Коли транзакція проходить успішно, платіжний шлюз самостійно ініціює фоновий запит (Webhook) на спеціальний прихований маршрут нашого сервера. Бекенд перевіряє справжність цього запиту за допомогою секретних ключів, знаходить відповідне замовлення в

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		49

базі даних і змінює його статус на оплачено, паралельно записуючи ідентифікатор банківської транзакції. Лише після цього система може відправити клієнту електронний лист із підтвердженням успішної покупки.

Модуль відгуків та зворотного зв'язку. Цей модуль забезпечує соціальний доказ якості кондитерських виробів. Його реалізація передбачає, що залишати коментарі можуть виключно авторизовані користувачі. Фронтенд відображає форму для відгуку лише тим клієнтам, які мають активний JWT-токен. При відправці форми сервер витягує ідентифікатор користувача з токена, бере ідентифікатор товару та текстовий коментар із запиту, і створює новий рядок у таблиці відгуків бази даних.

Приклад SQL-запиту для динамічного розрахунку середнього рейтингу товару:

```
SELECT  
  p.product_id,  
  p.name,  
  COUNT(r.review_id) as total_reviews,  
  ROUND(AVG(r.rating), 1) as average_rating -- Розрахунок середнього з  
округленням  
FROM products p  
LEFT JOIN reviews r ON p.product_id = r.product_id  
WHERE p.product_id = $1  
GROUP BY p.product_id;
```

Для відображення рейтингу на сторінці товару бекенд не зберігає середню оцінку як окреме статичне число. Замість цього, при завантаженні інформації про конкретний десерт, сервер виконує агрегуючий SQL-запит, який збирає всі оцінки цього товару з таблиці відгуків та динамічно вираховує середнє арифметичне значення. Такий підхід гарантує, що рейтинг завжди

					КвРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		50

залишатиметься актуальним і захищеним від розсинхронізації даних, навіть якщо адміністратор згодом видалить нецензурний або спамний коментар.

3.3 Реалізація інтерфейсу та інструкція користувача

Якщо клієнт хоче здійснити замовлення він заходить на ресурс і бачить там головну сторінку (рисунок 3.1). Загалом даний вебзастосунок має свої ключові особливості, а саме:

- колірну палітру, де використано гарні відтінки - кремовий, рожевий та глибокий шоколадний, що створює атмосферу затишку та преміальності;
- типографіку, що поєднує комбінацію елегантного шрифту із зарубками (Serif) для заголовків та чистого Sans-Serif для тексту, що забезпечує легкість читання.

Функціональність містить:

- фільтрацію, тобто користувач може швидко перемикатися між категоріями кондитерських виробів;
- інтерактивний кошик, де реалізовано бічну панель (drawer), яка дозволяє керувати покупками, не залишаючи сторінку;
- адаптивність, тобто дизайн повністю оптимізований для мобільних пристроїв та планшетів;
- UX-елементи, тобто картки товарів мають ефекти наведення, наприклад, зум фото, відображають рейтинг та ціну, що підвищує довіру клієнта.

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		51

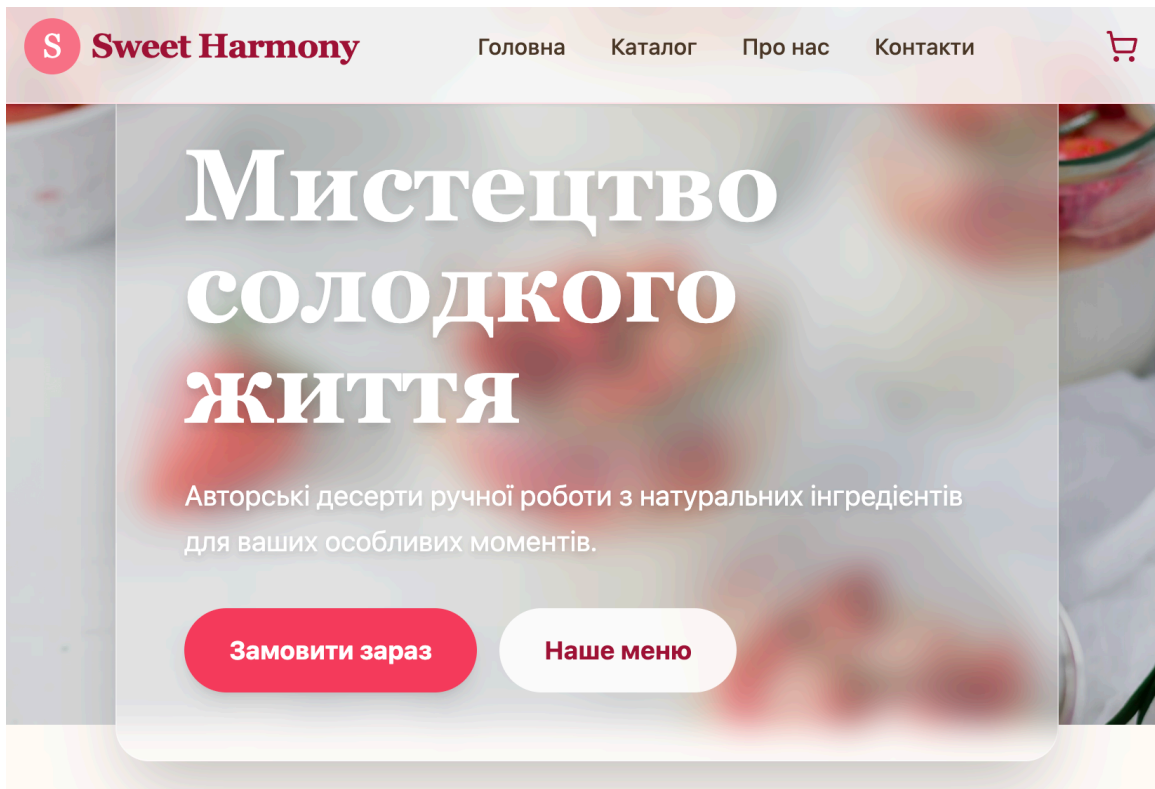


Рисунок 3.1 – Головна сторінка ресурсу для продажу кондитерських виробів

Для зручного користування є можливість реєстрації та авторизації. Так, на навігаційній панелі є кнопка *Увійти*, яка відкриває стильне модальне вікно з перемиканням між входом та створенням нового акаунта (рисунок 3.2).

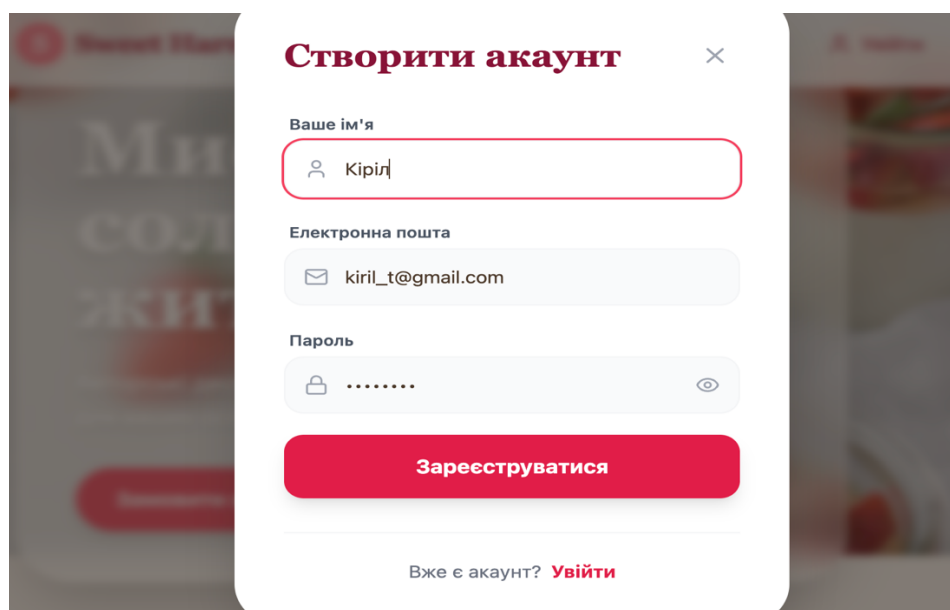


Рисунок 3.2 – Реєстрація користувача

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		52

Окрім візуальної частини, також додано стан *user*, який дозволяє імітувати вхід у систему, змінюючи інтерфейс через показ імені користувача замість кнопки входу (рисунок 3.3).

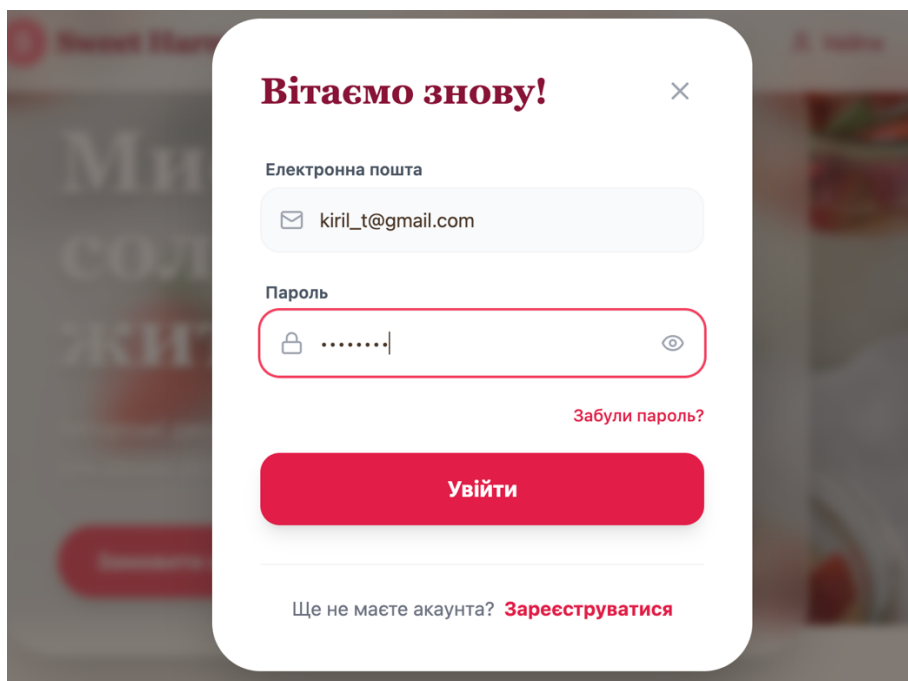


Рисунок 3.3 – Авторизація користувача

Також ресурс містить:

- модальне вікно, завдяки якому при натисканні на пункт *Увійти* з'являється елегантне спливаюче вікно з фоновим розмиттям;
- два режими, а саме легке перемикання між формою входу та реєстрації прямо в модальному вікні;
- покращена безпека, що видно візуально завдяки доданій можливості перегляду чи приховування пароля завдяки іконці ока;
- стан користувача, коли після входу кнопка імітує успішну дію замість кнопки входу у верхній панелі відображається ім'я користувача.
- дизайн форм, тобто поля введення мають плавні анімації, іконки та сучасний вигляд, що відповідає загальній естетиці сайту.

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
						53
Зм.	Арк	№ докум.	Підпис	Дата		

3.4 Тестування вебзастосунку

Тестування цієї системи побудоване на комплексному багаторівневому підході, який має гарантувати безперебійну роботу як зручного клієнтського інтерфейсу, так і складної прихованої серверної бізнес-логіки. Головною метою є виявлення проблем до того, як з ними зіткнеться реальний покупець під час замовлення десертів.

Першим етапом перевірки є модульне тестування, яке ізольовано перевіряє найдрібніші незалежні компоненти системи. На стороні фронтенду спеціальні автоматизовані скрипти перевіряють, чи правильно React-компоненти оновлюють загальну суму в кошику при зміні кількості обраних тістечок, не звертаючись при цьому до сервера. На серверному боці модульні тести підтверджують абсолютну коректність бізнес-логіки. Наприклад, вони перевіряють, що функція створення криптографічного хешу для пароля нового клієнта завжди повертає унікальне значення заданої довжини, а генератор токенів авторизації правильно встановлює час їхнього життя та коректно відхиляє прострочені ключі доступу.

Наступним критичним рівнем є інтеграційне тестування, яке повністю фокусується на правильній взаємодії нашої програми з базою даних та зовнішніми платіжними сервісами. Саме на цьому етапі ретельно тестуються складні SQL-запити. Для цього розгортається окрема тестова база даних, яка наповнюється десятками фіктивних відгуків про товари, після чого автоматизований тест виконує реальний агрегуючий запит і перевіряє, чи правильно функції об'єднання таблиць та математичного усереднення повертають очікуваний рейтинг десерту. Також тут жорстко перевіряється надійність транзакцій при оформленні замовлення: тест навмисно імітує раптовий збій на етапі запису деталей чека, щоб на сто відсотків переконатися, що база даних успішно скасовує всі попередні дії та не списує товар зі складу помилково.

					КвРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		54

Найвищим рівнем перевірки є наскрізне тестування, яке за допомогою спеціальних програм повністю симулює поведінку реального клієнта безпосередньо у вікні браузера. Автоматизовані роботи проходять весь шлях звичайного користувача: вони відкривають сторінку каталогу, застосовують фільтри для пошуку макаронів, додають обраний набір до кошика, переходять до екрана оформлення замовлення, заповнюють форму доставки реальними тестовими даними та імітують натискання кнопки оплати. Цей підхід гарантує, що всі розрізнені модулі платформи злагоджено працюють разом як єдиний механізм, і жодне випадкове оновлення коду розробниками не зламає ключовий процес здійснення покупки.

Окрім функціональних перевірок, система обов'язково проходить стресове та навантажувальне тестування. Спеціальні утиліти генерують штучний наплив тисяч одночасних віртуальних відвідувачів, які одночасно переглядають каталог та додають товари в кошик. Це робиться для імітації пікових навантажень напередодні великих свят, таких як День святого Валентина або Новий рік. Таке тестування дозволяє перевірити, чи витримає хмарна інфраструктура та база даних такий обсяг трафіку без критичного уповільнення відображення вітрини та без відмов у прийнятті замовлень.

У таблицях подано тестові сценарії для перевірки працездатності головних модулів платформи. Кожен сценарій описує початкові умови, послідовність дій та очікувану реакцію системи і бази даних. Формат таблиць дозволяє QA-інженерам чітко фіксувати результати проходження тестів.

У таблиці 3.1 подано приклад тестового сценарію, за допомогою якого перевіряється безпека доступу та коректність обробки персональних даних клієнтів.

Таблиця 3.1 – Тестовий сценарій перевірки модуля авторизації та управління профілем

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		55

Назва сценарію	Передумови	Кроки виконання	Очікуваний результат
Успішна авторизація	Користувач вже має обліковий запис у базі даних	<ol style="list-style-type: none"> 1. Відкрити форму входу. 2. Ввести валідний email та пароль. 3. Натиснути кнопку <i>Увійти</i>. 	Система генерує JWT-токен, зберігає його в браузері. Модальне вікно закривається, в навігації з'являється ім'я користувача.
Реєстрація з існуючим email	Відкрита форма реєстрації нового клієнта	<ol style="list-style-type: none"> 1. Ввести електронну пошту, яка вже прив'язана до іншого акаунта. 2. Заповнити інші поля. 3. Відправити форму. 	Сервер відхиляє запит через обмеження UNIQUE у БД. Клієнт бачить повідомлення <i>Користувач з такою поштою вже існує</i> .
Захист закритих маршрутів	Користувач не авторизований (немає токена)	<ol style="list-style-type: none"> 1. Спробувати надіслати прямиї API-запит до маршруту <code>/api/user/orders</code> (Історія замовлень). 	Сервер повертає помилку 401 <i>Unauthorized</i> . Доступ до бази даних блокується.

У таблиці 3.2 подано тестовий сценарій, за допомогою якого тестується клієнтська логіка взаємодії з товарами, перерахунок вартості та реакція інтерфейсу.

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		56

Таблиця 3.2 – Тестування модуля каталог та Кошик

Назва сценарію	Передумови	Кроки виконання	Очікуваний результат
Додавання товару	Кошик порожній, користувач знаходиться на сторінці каталогу	1. Натиснути кнопку <i>Додати в кошик</i> на картці <i>Малиновий оксамит</i> . 2. Відкрити бічну панель кошика.	Товар відображається у висувній панелі. Іконка в навігації показує цифру 1. Загальна сума дорівнює ціні товару.
Перерахунок суми	У кошику знаходиться один товар	1. Відкрити кошик. 2. Збільшити кількість товару, натиснувши на відповідну кнопку (+).	Кількість одиниць збільшується. Загальна сума замовлення миттєво перераховується без перезавантаження сторінки.
Фільтрація каталогу	Користувач на сторінці каталогу зі всіма товарами	1. Натиснути на кнопку-фільтр <i>Макарони у</i> верхній панелі.	Відображаються лише товари з відповідною категорією. Решта карток плавно зникає з екрана.

У таблиці 3.3 показано тестування найбільш критичного модуля, де перевіряється цілісність транзакцій бази даних та взаємодія з фінансами.

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		57

Таблиця 3.3 – Тестування модуля оформлення замовлення

Назва сценарію	Передумови	Кроки виконання	Очікуваний результат
Успішне оформлення	У кошику є товари, клієнт авторизований	1. Натиснути <i>Оформити замовлення</i> . 2. Заповнити адресу доставки. 3. Підтвердити операцію.	У БД створюється запис у таблицях <i>orders</i> та <i>order_items</i> . Статус замовлення - <i>Pending</i> . Залишки на складі оновлюються.
Недостатньо товару на складі	Товар <i>Лимонна тарталетка</i> має залишок 1 шт. у базі даних	1. Додати в кошик 2 шт. цього товару. 2. Спробувати оформити замовлення.	Сервер перериває транзакцію (Rollback). Користувач отримує сповіщення, що потрібної кількості немає в наявності.
Успішний колбек від банку	Замовлення створено (статус Pending)	1. Зімітувати відправку успішного Webhook-запиту від платіжного шлюзу на сервер.	Статус замовлення в базі даних автоматично змінюється на <i>Paid</i> . В поле <i>payment_id</i> записується номер транзакції.

Зм.	Арк	№ докум.	Підпис	Дата

За результатами проведеного тестування, яке охопило ключові модулі системи від етапу авторизації до фінального оформлення замовлення, можна констатувати високий рівень стабільності та повної готовності платформи до експлуатації. Усі критичні шляхи користувача функціонують безвідмовно, забезпечуючи плавний, швидкий та інтуїтивно зрозумілий перехід між вибором десерту, роботою з кошиком та проведенням оплати.

Особливу увагу під час перевірки було приділено цілісності даних та безпеці транзакцій на рівні бази даних PostgreSQL. Тестові сценарії підтвердили, що бекенд коректно обробляє конфліктні ситуації. Наприклад, при спробі замовити товар, якого вже немає на складі, система миттєво спрацьовує на захист, надійно скасовуючи операцію без будь-яких фінансових чи облікових втрат. Інтеграція з платіжним шлюзом також продемонструвала свою надійність: система безпомилково розпізнає успішні транзакції та автоматично оновлює статуси замовлень через захищені вебхуки без втручання адміністратора.

Підсумовуючи, архітектура застосунку довела свою стійкість як на рівні зручного клієнтського React-інтерфейсу, так і на рівні складної серверної логіки. Система захищена від несанкціонованого доступу, стійка до помилок введення даних та готова витримувати реальні навантаження. Вебзастосунок повністю відповідає заявленим вимогам щодо безпеки, продуктивності та сучасного користувацького досвіду, що дозволяє впевнено рекомендувати його до запуску в промислове середовище.

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
						59
Зм.	Арк	№ докум.	Підпис	Дата		

ВИСНОВКИ

Отже, під час написання кваліфікаційної роботи було виконано повний цикл завдань, пов'язаних із проєктуванням та розробкою вебзастосунку для електронної комерції у кондитерській галузі. Проведена робота дозволила закріпити теоретичні знання та здобути практичні навички створення сучасних програмних продуктів.

За результатами роботи можна зробити висновки про вирішення поставлених задач, а саме:

- вирішення реальної бізнес-задачі;
- опанування сучасного технологічного стеку;
- інтеграція зовнішніх сервісів;
- забезпечення якості та безпеки;
- проєктування користувацького досвіду.

На основі аналізу предметної області доведено, що автоматизація процесу замовлення кастомних десертів через вебзастосунок вирішує головні проблеми галузі: зменшує час обробки замовлень, усуває помилки людського фактора та гарантує фінансову безпеку через онлайн-передоплату.

У процесі розробки успішно застосовано архітектурні патерни для створення REST API та реляційної бази даних. Налаштування середовища розробки, зокрема віртуалізація через Docker та використання систем контролю версій Git здійснювалося з використанням наданої матеріально-технічної бази.

Здобуто практичний досвід інтеграції платіжних шлюзів, наприклад, Monopray та налаштування механізмів обробки Webhooks, що є критично важливою навичкою для розробки будь-яких e-commerce рішень.

Дана робота дозволила заглибитися у процеси забезпечення якості коду через Code Review та базове автоматизоване тестування (Unit-тести).

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		60

Розроблено логіку інтерактивного конструктора десертів, що дозволяє конвертувати складні візуальні побажання клієнта у структурований формат даних JSON для зручної обробки бекендом та передачі на виробництво (кухню).

Мета роботи досягнута в повному обсязі. Розроблений прототип вебзастосунку має високий рівень актуальності, відповідає сучасним вимогам FoodTech-індустрії та може бути використаний як основа для впровадження в реальне комерційне середовище. Здобуті компетенції повністю відповідають кваліфікаційним вимогам до розробника програмного забезпечення.

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		61

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Кваліфікаційна робота : Методичні настанови для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 121 «Інженерія програмного забезпечення» / Л. П. Бедратюк, Г. І. Радельчук. Хмельницький : ХНУ, 2023. 60 с.
2. Коуторн М. Створення високоефективних вебсайтів. Львів : Видавництво Старого Лева, 2022. 340 с.
3. Фленова О. В. Проектування інтерфейсів для систем електронної комерції : навч. посіб. Харків : ХНУРЕ, 2023. 156 с.
4. Теорія та практика розробки веб-застосунків на мові JavaScript : монографія / А. Б. Іванов та ін. Київ : Політехніка, 2021. 288 с.
5. Швайка Л. А. Економіка та організація торгівлі : навч. посіб. Київ : Знання, 2018. 320 с.
6. React Documentation : офіційна документація. URL: <https://react.dev> (дата звернення: 22.02.2026).
7. PostgreSQL 15 Documentation : офіційна документація. URL: <https://www.postgresql.org/docs/15/index.html> (дата звернення: 22.02.2026).
8. ДСТУ 4135:2021. Вироби кондитерські. Цукерки. Загальні технічні умови. [Чинний від 2022-01-01]. Київ : ДП «УкрНДНЦ», 2021. 24 с.
9. Гамма Е., Гелм Р., Джонсон Р., Вліссідес Дж. Прийоми об'єктно-орієнтованого проектування. Патерни проектування. Київ : Діалектика, 2020. 368 с.
10. Кульков С. О. Основи безпеки веб-застосунків та захисту даних в e-commerce. *Комп'ютерні науки та інженерія*. 2024. № 4. С. 45–52.
11. Наскрізна практична підготовка: програма та методичні вказівки щодо її організації та виконання студентами спеціальності 121 «Інженерія програмного забезпечення» освітнього ступеня «бакалавр» / Г. І. Радельчук, Л. П. Бедратюк, – Хмельницький : ХНУ, 2021. – 40 с.

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		62

12. Про електронну комерцію : Закон України від 03.09.2015 № 675-VIII.
URL: <https://zakon.rada.gov.ua/laws/show/675-19> (дата звернення: 22.02.2026).

13. Ларман К. Застосування UML та шаблонів проєктування : підручник.
Київ : Діалектика, 2019. 736 с.

14. Дорош О. С. Технології розробки клієнтської частини веб-застосунків (Frontend Development) : навч. посіб. Тернопіль : ТНТУ, 2023. 215 с.

15. Марченко В. П. Оптимізація конверсії в інтернет-магазинах продуктів харчування та кондитерських виробів. *Менеджмент та бізнес*. 2024. № 2. С. 34-41.

16. Бакстон Б. Досвід користувача: проєктування взаємодії та дизайн інтерфейсів. Київ : Основи, 2021. 312 с.

17. Model View ViewModel: Usage of MVVM in Muvi's App Architecture.
URL: <https://www.muvi.com/blogs/model-view-viewmodel-mvvm/> (дата звернення 12.04.2026).

18. Клієнт-серверна архітектура. URL: <https://javarush.com/ua/quests/lectures/ua.questservlets.level14.lecture00> (дата звернення 12.04.2026).

19. Кріспін Л., Грегори Д. Гнучке тестування: практичне керівництво для тестувальників та QA-інженерів. Харків : Гриф, 2022. 464 с.

20. Дмитренко М. В. Організація автоматизованого модульного тестування вебдодатків за допомогою Jest. *Системне програмування та IT-технології*. 2024. № 1. С. 58–64.

21. Stripe API Reference Documentation. URL: <https://stripe.com/docs/api> (дата звернення: 25.02.2026).

22. Документація послуги Monopay (від Monobank) для розробників. URL: <https://api.monobank.ua/docs/> (дата звернення: 27.02.2026).

23. Сидоренко В. П., Ткаченко М. О. Застосування UUID як первинних ключів у розподілених базах даних e-commerce платформ. *Комп'ютерні системи та мережі*. 2024. № 8. С. 73–80.

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		63

24. Редіс К., Сегера М. Розробка високонавантажених систем із використанням кешування в Redis. Київ : Максимум, 2023. 195 с.

25. Тейлор Б. Оптимізація складних SQL-запитів у реляційній СКБД PostgreSQL. Харків : Факт, 2022. 210 с.

26. Ньюмен С. Створення мікросервісів: проектування тонко налаштованих систем. Львів : Видавництво Старого Лева, 2022. 416 с.

27. Річардсон К. Патерни мікросервісної архітектури. Київ : Діалектика, 2021. 544 с.

28. Беррес Ф. Ефективний TypeScript: 62 конкретні способи покращити ваш код. Київ : Наукова думка, 2023. 280 с.

29. Wieruch R. The Road to Next.js: Your journey to master Next.js in React. Berlin : Independent Publisher, 2023. 310 p.

30. Дакетт Д. HTML і CSS. Розробка та дизайн вебсайтів. Львів : Видавництво Старого Лева, 2021. 480 с.

					КВРІПЗ.2201112.01.17.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		64

Додаток А
(обов'язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вебзастосунок для продажу кондитерських виробів

Здобувач ступеня бакалавра:

Трембач Кіріл Олегович

гр. ІПЗ-22-1

Керівник роботи:

Яшина Оксана Миколаївна

канд. техн. наук, доцент

Рисунок А.1 – Титульний слайд, представлення

Мета та завдання роботи

Мета роботи:

розробка вебзастосунку для продажу кондитерських виробів, що містить функціональні можливості для купівлі солодошів, користуючись мережею Інтернет.

Завдання кваліфікаційної роботи:

- Здійснити аналіз ринку
- Проаналізувати існуючі програмні рішення
- Встановити вимоги до розробки ПЗ
- Встановити вимоги до ТЗ
- Проаналізувати та обрати тип архітектури
- Провести реалізацію
- Здійснити тестування

Рисунок А.2 – Мета та завдання кваліфікаційної роботи

Актуальність теми



Купівля "очима"

Якісний вебзастосунок дозволяє створити ефект присутності та викликає негайне бажання спробувати продукт.



Швидкість вибору

У світі, де час є найціннішим ресурсом, можливість зробити замовлення в три кліки через смартфон стає вирішальним фактором.

Інтеграція онлайн-оплати

Мінімізує ризик «незабраних» замовлень, а також пришвидшує процес обробки замовлення.

Рисунок А.3 – Актуальність теми

Предметна область

Предметна область охоплює процеси виробництва, складського обліку, логістики та маркетингу десертів, а саме:

- Гіпер-персоналізація (конструктор тортів);
- Омніканальність (Instagram/TikTok/вебсайт);
- Прозорість складу(Система повинна точно знати залишки сировини)



Рисунок А.4 – Предметна область

Аналіз стану існуючих рішень

Характеристика	Маркетплейси	Брендові застосунки	Конструктори(SaaS)
UX/UI дизайн	Стандартний	Унікальний	Шаблонний
Кастомізація товару	Відсутня	Висока	Низька
Швидкість розробки	Миттєво(реєстрація)	Довго(3-6 місяців)	Швидко(1-2 тижні)
Вартість володіння	Комісія з продажів	Висока(підтримка)	Абонплата

Рисунок А.5 – Аналіз стану існуючих рішень

Вимоги до програмного забезпечення

Функціональні:

- Надавати зручний каталог товарів із можливістю фільтрації за категоріями;
- Надати змогу гнучко змінювати кількість позицій;
- Можливість додати кастомні побажання;
- Система має підтримувати реєстрацію та авторизацію;

Нефункціональні:

- Інтерфейс має проектуватися за строгим принципом Mobile-First;
- Час завантаження сторінок має бути мінімальним;
- Використання захищеного протоколу HTTPS;
- Архітектура повинна бути надійною та масштабованою;

Рисунок А.6 – Аналіз вимог

Діаграма варіантів використання

У даній діаграмі такі дійові особи:

- **Зареєстрований клієнт** – основний користувач, який має доступ до розширеного функціоналу;
- **Адміністратор**, що отримує повідомлення та обробляє замовлення, створені клієнтом;
- **Система оплати** або зовнішній сервіс, наприклад, LiqPay або Stripe, що взаємодіє з процесом оплати;

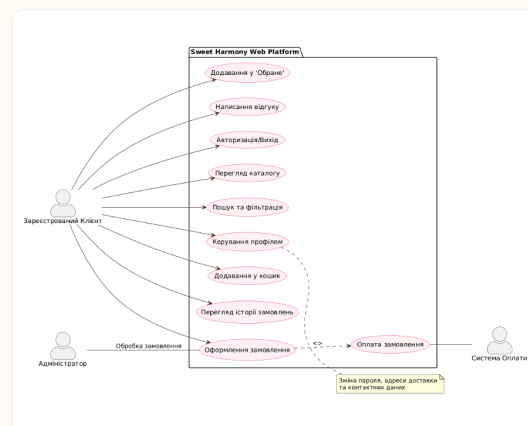


Рисунок А.7 – Діаграма варіантів використання

Діаграма варіантів використання

У даній діаграмі такі дійові особи:

- **Зареєстрований клієнт** – основний користувач, який має доступ до розширеного функціоналу;
- **Адміністратор**, що отримує повідомлення та обробляє замовлення, створені клієнтом;
- **Система оплати** або зовнішній сервіс, наприклад, LiqPay або Stripe, що взаємодіє з процесом оплати;

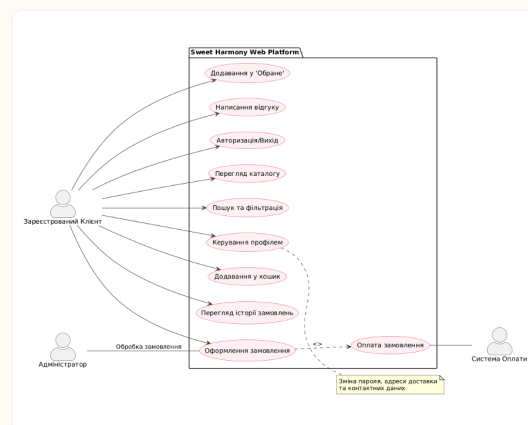


Рисунок А.8 – Опис декомпозиції, залежностей, інтерфейсів

Проектування даних

- Для розробки бази даних було обрано реляційну модель даних.
- Логічне проектування полягало в розробці структури таблиць та зв'язків між ними.
- База даних містить інформацію про користувачів, товари, замовлення, категорії та відгуки.
- Всі таблиці були нормалізовані до третьої нормальної форми для забезпечення цілісності даних та уникнення дублювання.
- Зв'язки між таблицями реалізовано за допомогою первинних та зовнішніх ключів.

Рисунок А.9 – Проектування модулів і даних

Вибір технологій

React.js

Це ідеальний вибір для електронної комерції завдяки технології Server-Side Rendering (SSR), яка забезпечує відмінну SEO-оптимізацію

Node.js

Використовується в якості швидкого та масштабованого середовища для створення API

PostgreSQL

Основна реляційна база даних для фіксації замовлень, профілів клієнтів та аналітики

Стек обрано для забезпечення масштабованості, швидкості та сучасного UI.

Рисунок А.10 – Вибір технологій

Реалізація модулів і база даних



Рисунок А.11 – Реалізація модулів і база даних

Результати: Реєстрація та авторизація

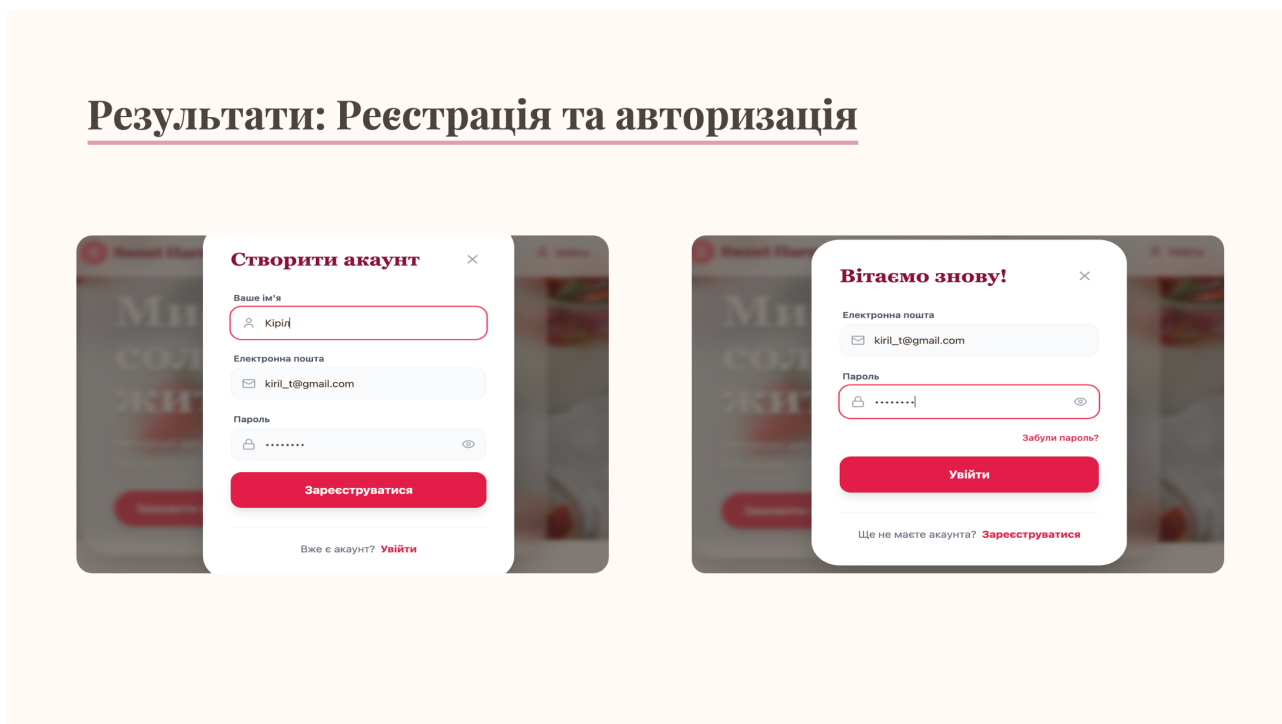


Рисунок А.12 – Результати: Реєстрація та авторизація

Вимоги до забезпечення

Технічне забезпечення

Мінімальні вимоги:

- процесор: Pentium 4 з SSE2;
- вінчестер: 350 Mb;
- відеоадаптер: 3D адаптер nVidia;
- оперативна пам'ять: 512 Mb;
- Аудіокарта: будь-яка звукова плата.
- роздільна здатність екрану: SVGA 800x600
- контролери: клавіатура, миша;

Програмне забезпечення

Мінімальні вимоги:

- операційна система: Windows 7;
- розрядність: x86 або x64;

Рисунок А.13 – Вимоги до технічного та програмного забезпечення

Тестування ПЗ

За результатами проведеного тестування, яке охопило ключові модулі системи від етапу авторизації до фінального оформлення замовлення, можна констатувати високий рівень стабільності та повної готовності платформи до експлуатації.

Назва сценарію	Передумови	Кроки виконання	Очікуваний результат
Успішне оформлення	У кошику є товари, клієнт авторизований	1. Натиснути Оформити замовлення. 2. Заповнити адресу доставки. 3. Підтвердити операцію.	У БД створюється запис у таблицях orders та order_items. Статус замовлення - Pending. Залишки на складі оновлюються.
Недостатньо товару на складі	Товар Лимонна таргалетка має залишок 1 шт. у базі даних	1. Додати в кошик 2 шт. цього товару. 2. Спробувати оформити замовлення.	Сервер перериває транзакцію (Rollback). Користувач отримує сповіщення, що потрібної кількості немає в наявності.
Успішний колбек від банку	Замовлення створено (статус Pending)	1. Змітувати відправку успішного Webhook-запиту від платіжного шлюзу на сервер.	Статус замовлення в базі даних автоматично змінюється на Paid. В поле payment_id записується номер транзакції.

Рисунок А.14 – Тестування ПЗ

Висновки

Як результат маємо виконаний повний цикл завдань, пов'язаних із проектуванням та розробкою вебзастосунку для електронної комерції. Також вирішені бізнес-задачі, а саме автоматизація зменшує час обробки замовлень та гарантує фінансову безпеку через онлайн-передоплату. Що до технологічних досягнень, то успішно було застосовано архітектурні патерни для створення REST API та реляційної бази даних.

Як ключовий функціонал було розроблено логіку конструктора десертів, що конвертує побажання клієнта у структурований формат JSON.

У підсумку – мета роботи досягнута в повному обсязі; розроблений прототип відповідає сучасним вимогам FoodTech-індустрії.

Рисунок А.15 – Висновки

Дякую за увагу!

Готовий до ваших запитань

Трембач Кіріл Олегович

гр. ІПЗ-22-1



Рисунок А.16 – Останній слайд

Додаток Б
(обов'язковий)

ТЕХНІЧНЕ ЗАВДАННЯ

Вступ

Робота виконується в рамках кваліфікаційної роботи на тему «Вебзастосунок для продажу кондитерських виробів».

1 Підстава для розробки

Підставою для розробки даного програмного забезпечення є «Завдання на кваліфікаційну роботу», що затверджене завідувачем кафедри інженерії програмного забезпечення.

Найменування розробки: «Вебзастосунок для продажу кондитерських виробів».

2 Призначення розробки

Вебзастосунок для продажу кондитерських виробів призначений для забезпечення користувачів можливістю швидкого пошуку, огляду та купівлі кондитерських товарів в будь-якій точці світу та з будь-якої точки світу, не переживаючи за свою безпеку та безпеку своїх даних.

Користувачами системи є звичайні люди, що вирішили купити солодощі. Пошук необхідного товару здійснюється шляхом вказання безпосередньої адреси ресурсу, коли SEO спрацювала добре та ресурс достатньо знає людей або ж коли вказуються параметри.

Також передбачено форму для зворотного зв'язку, можливість добавляти товар у корзину, здійснювати замовлення, обирати спосіб оплати та доставки

Щодо експлуатаційного призначення, система може використовуватися на будь-якому персональному комп'ютері, у будь-якому встановленому браузері, ніяких додаткових налаштувань не потребує.

3 Вимоги до програми

3.1 Вимоги до функціональних характеристик

Онлайн-ресурс з продажу кондитерських виробів повинен мати можливість:

– публікації, редагування та видалення інформації щодо продукції, а саме солодошів;

– зручного пошуку у категоріях продукції;

– авторизації та реєстрації користувачів;

– налаштування особистого кабінету та деактивації профілю;

– керування онлайн-ресурсом адміністратором;

Вимоги до інтерфейсу:

– простий, сучасний та зрозумілий інтерфейс користувача;

– актуальна інформація щодо товарів.

3.2 Вимоги до надійності

Розроблюване ПЗ повинно мати:

– можливість самовідновлення після збоїв, таких як відключення електроживлення;

– обмеження несанкціонованого доступу до даних;

– розмежування прав користувачів у системі;

– контроль інформації, що вводиться, та блокування некоректних дій користувача при роботі з системою.

3.3 Вимоги до складу та параметрів технічних засобів

Мінімальні вимоги для функціонування системи повинні відповідати вимогам будь-якого браузеру. Нижче наведено приклад мінімальних вимог для браузера Google Chrome:

– стабільне з'єднання з Інтернетом;

– операційна система: Windows 10, 8, 8.1, 7;

– розрядність: x86 (32-bit) або x64 (64-bit);

– процесор: Pentium 4 з SSE2;

– вінчестер: 350 Mb;

– відеоадаптер: 3D адаптер nVidia, Intel, AMD / ATI;

– оперативна пам'ять: 512 Mb;

– будь-яка аудіокарта;

- роздільна здатність екрану: SVGA 800x600;
- контролери: клавіатура, миша.

3.4 Вимоги до інформаційної та програмної сумісності

Для створення серверної частини будуть використовуватися такі технології:

Для створення клієнтської частини використовуватимуться такі технології:

- мова розмітки HTML;
- таблиці стилів CSS;
- мова програмування JavaScript.

Для роботи з базою даних буде використовуватися мова SQL та система керування базами даних SQL Server.

3.5 Вимоги до транспортування та зберігання

Програма, її документація поставляються у цифровому вигляді. Умови експлуатації програмного забезпечення збігаються з умовами експлуатації серверу, на якому буде розміщене ПЗ.

3.6 Спеціальні вимоги

Програма повинна мати дружній інтерфейс, розрахований на користувача середньої кваліфікації з точки зору комп'ютерної грамотності. Мова програмування визначається вибором виконавця.

4 Вимоги до програмної документації

В ході розробки програми повинні бути підготовлені:

- технічне завдання;
- опис програми;
- текст програми з коментарями та поясненнями;
- методика випробувань;
- відомості про функціонування програми;
- керівництво користувача;
- керівництво програміста.

5 Стадії та етапи розробки

Розробка програмного продукту проходить декілька стадій і етапів:

- розробка технічного завдання;

- створення ескізного проекту;
- створення технічного проекту;
- створення робочого проекту;
- розробка програмної документації;
- проведення тестування онлайн-ресурсу;
- впровадження.

6 Порядок контролю та приймання

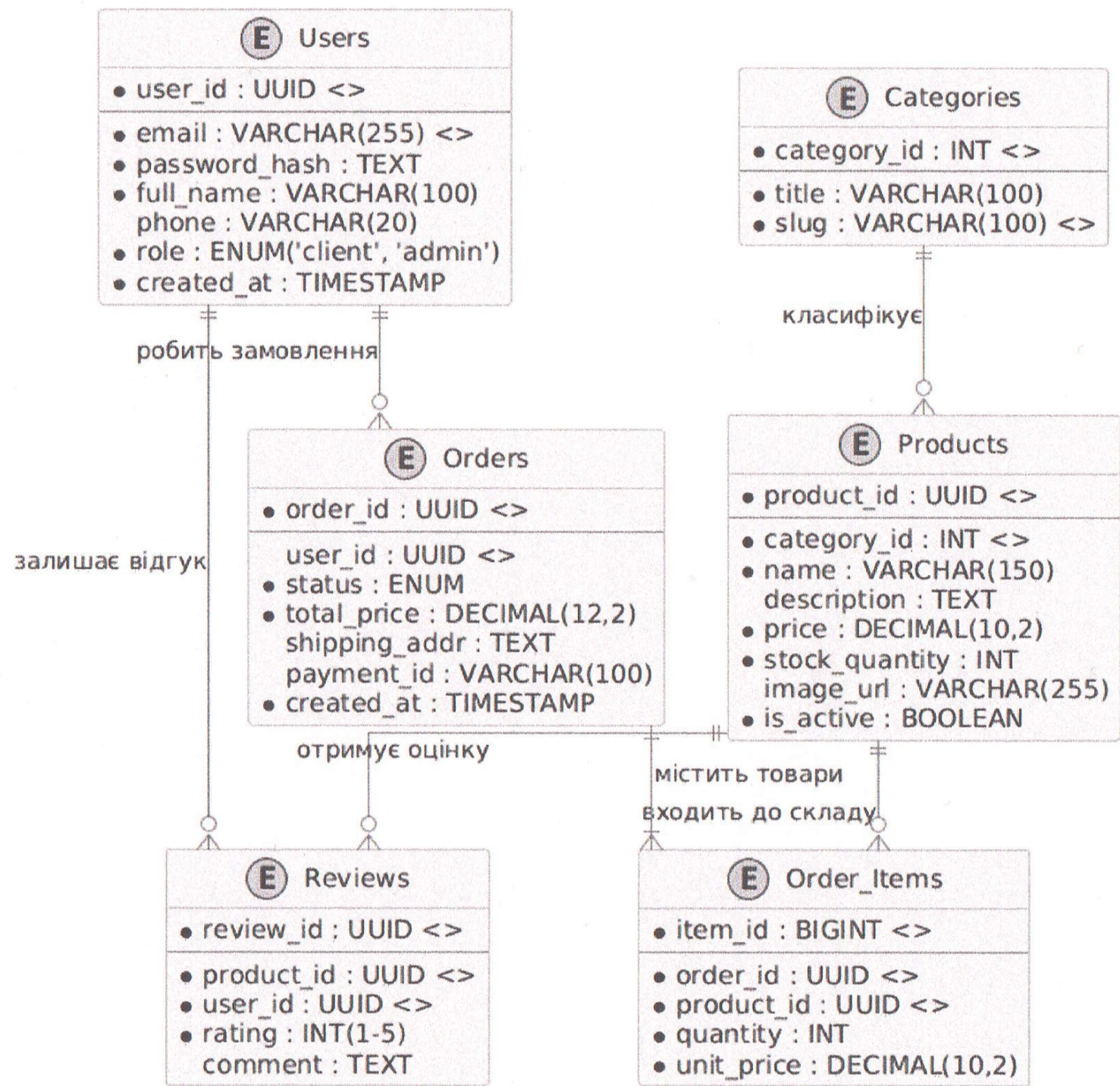
Контроль здійснюється кінцевими користувачами онлайн-ресурсу, підключеними на етапі тестування. Прийом онлайн-ресурсу здійснюється після його повного розгортання та встановлення на хостинг і налаштування для нормального функціонування.

Після закінчення розробки онлайн-ресурсу повинні бути проведені тестування на захист від некоректного введення.

ГРАФІЧНА ЧАСТИНА

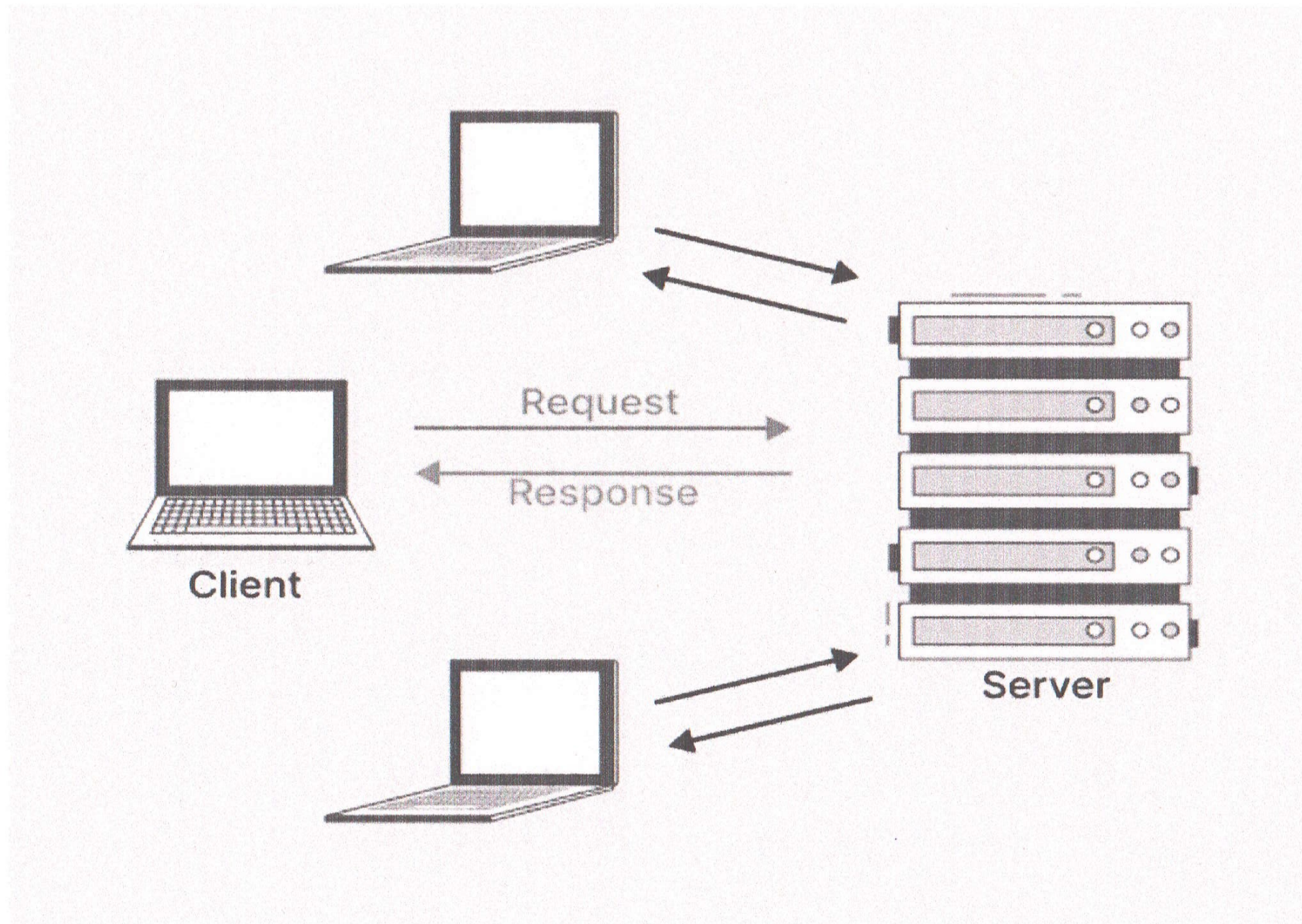


						КвРІПЗ.2201112.01.17.ПЗ		
						Вебзастосунок для продажу кондитерських виробів		
Зм.	Арк.	№ докум.	Підпис	Дата				
Розробив	Трембач К.О.		<i>[Signature]</i>	29.05	Літера	Маса	Масштаб	
Керівник	Яшина О.М.		<i>[Signature]</i>	29.05	Діаграма варіантів використання			
Консульта.								
Н. Контр.	Праворська Н.Г.		<i>[Signature]</i>	29.05	Аркуш 1	Аркушів 3		
Зав. каф.	Бедратюк Л. П.		<i>[Signature]</i>	29.05	ХНУ, ІПЗ-22-1			



				КвРІПЗ.2201112.01.17.ПЗ		
				Вебзастосунок для продажу кондитерських виробів		
				Схема бази даних		
				Літера		Маса
				Масштаб		
				Аркуш 2		Аркушів 3
				ХНУ, ІПЗ-22-1		

Зм.	Арк.	№ докум.	Підпис	Дата
Розробник		Трембач К.О.	<i>[Signature]</i>	29.06
Керівник		Яшина О.М.	<i>[Signature]</i>	29.05
Консульт.				
Н. Контр.		Праворська Н.І.	<i>[Signature]</i>	29.06
Зав. каф.		Бедратюк Л. П.	<i>[Signature]</i>	29.05



					КвРІПЗ.2201112.01.17.ПЗ					
					Вебзастосунок для продажу кондитерських виробів			Літера	Маса	Масштаб
Зм.	Арк.	№ докум.	Підпис	Дата	Архітектура					
Розробив		Трембач К.О.	<i>[Signature]</i>	29.05						
Керівник		Яшина О.М.	<i>[Signature]</i>	29.05				Аркуш 3	Аркушів 3	
Консульт.										
Н. Контр.		Праворська Н.І.	<i>[Signature]</i>	29.05				ХНУ, ІПЗ-22-1		
Зав. каф.		Бедратюк Л. П.	<i>[Signature]</i>	29.05						

СУПРОВІДНІ МАТЕРІАЛИ

Завідувачу кафедри інженерії програмного
забезпечення проф. Леоніду БЕДРАТЮКУ
здобувача вищої освіти
Трембача Кіріла Олеговича
факультет ІТ, ІVкурс, група ІПЗ-22-1

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності в Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії Хмельницького національного університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-обчислювального комплексу StrikePlagiarism та/або програмно-технічного засобу AntiPlagiarism і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення текстових збігів у роботах.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

01.05.2026

дата


підпис

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Кіріл ТРЕМБАЧ

Співавтор:

Назва: Вебзастосунок для продажу кондитерських виробів

Науковий керівник: канд. техн. наук, доцент Оксана ЯШИНА

Підрозділ: Кафедра інженерії програмного забезпечення

Коефіцієнт подібності 1: 4.95%

Коефіцієнт подібності 2: 0%

Мікропробіли: 12

Заміна букв: 2

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2026-05-25 01:45:41.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

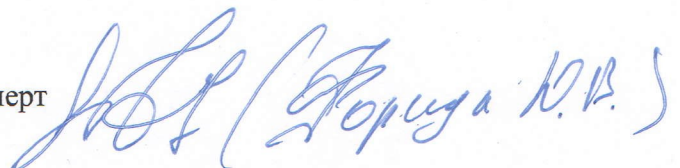
Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

Дата

25.05.2026

експерт

 (Яшина О.В.)



Anti-Plagiarism (<http://ap.km.ua>) v-16.718

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: UA, US, RU. Помилки в документах: 13%

ID: 272081 Назва: Вебзастосунок для продажу кондитерських виробів Додано в БД: 2026-05-25 Автора: Кіріл ТРЕМБАЧ Керівники: канд. техн. наук, доцент Оксана ЯШИНА Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	68896	565	3143 (5%)	39 (7%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «Бакалавр»

Дипломник Трембач Кіріл Олегович

Тема Вебзастосунок для продажу кондитерських виробів

Спеціальність 121 – Інженерія програмного забезпечення

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3; кількість сторінок записки 81

1. Короткий зміст пояснювальної записки та прийнятих рішень у кваліфікаційній роботі досліджено і проаналізовано предметну область реалізації кондитерських виробів, визначено усі функціональні та нефункціональні вимоги до вебзастосунку. Був проведений аналіз існуючих програмних рішень на ринку e-commerce, розглянуто їх переваги і недоліки, та доведено актуальність розробки нового вебзастосунку для продажу солодощів. Розглянуто сучасні інструменти й технології для реалізації спроектованих рішень, в результаті чого створено готовий програмний продукт. Також було проведено комплексне тестування вебзастосунку, за результатами якого доведено, що розроблене програмне забезпечення працює коректно та готове до експлуатації.

2. Висновок про відповідність роботи поставленому завданню Кваліфікаційна робота виконана відповідно до поставленого завдання та з дотриманням всіх вимог.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи у вступі доведено актуальність теми в рамках сучасного FoodTech-сегменту, визначено мету та завдання кваліфікаційного проектування. У першому розділі проведено аналіз предметної області електронної комерції у кондитерській галузі, розглянуто наявні ринкові рішення та визначені чіткі функціональні й нефункціональні вимоги до вебзастосунку. У другому розділі проведено аналіз сучасних підходів, на основі чого для системи обрано клієнт-серверну модель, шаблони MVVM та Clean Architecture, а також обґрунтовано стек технологій. У третьому розділі виконано послідовне розгортання реляційної бази даних, програмну реалізацію основних модулів із підключенням онлайн-оплат та розроблено інтерфейс користувача. Також у цьому розділі проведено комплексне багаторівневе тестування системи відповідно до технічних вимог, що підтвердило стабільність і коректність роботи створеного продукту.

4. Позитивні сторони роботи Тематика кваліфікаційної роботи є актуальною, оскільки на сьогодні в Україні спеціалізовані онлайн-платформи для кондитерської галузі потребують вищого рівня автоматизації та розширення функціональних можливостей для кастомізації замовлень. Також було застосовано новітні технології для побудови програмного продукту та актуальні архітектурні рішення.

5. Негативні сторони роботи У роботі фільтрація та пошук десертів були реалізовані переважно за категоріями — було б доцільно додати розширений пошук за конкретними інгредієнтами чи калорійністю. Також було б краще автоматично надсилати SMS-сповіщення або повідомлення у месенджери безпосередньо клієнту в момент, коли адміністратор змінює статус його замовлення на «Готово до доставки»

6. Оцінка графічного оформлення та пояснювальної Графічне оформлення виконано відповідно до теми кваліфікаційної роботи та подано у вигляді діаграм і рисунків. Пояснювальна записка оформлена згідно вимог чинних стандартів.

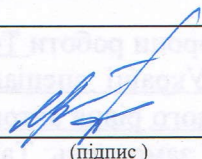
7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота заслуговує на позитивну оцінку. Матеріал пояснювальної записки структурований, послідовний, чіткий та простий, що дозволяє чітко зрозуміти викладений матеріал у рамках тематики проектування вебресурсу для кондитерської галузі. Графічний матеріал дає можливість наочно побачити деталі архітектури та бази даних розробленого вебзастосування.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана у повному обсязі, відповідає поставленій задачі та заслуговує на оцінку «добре».

РЕЦЕНЗЕНТ (прізвище, ім'я, по-батькові, посада, місце роботи) Капустян
Марія Вікторівна, кандидат технічних наук,
доцент за каф систем і захисту інформації,
ДНУ

“29” травня 2026 р.


(підпис)

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатами звіту/звітів перевірки роботи, продуктованими програмно-технічним засобом (ами), на наявність текстових збігів.

Назва кваліфікаційної роботи: «Вебзастосунок для продажу кондитерських виробів»

Автор: Трембач Кіріл Олегович

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Спеціальність: 121 – Інженерія програмного забезпечення

Науковий керівник: Яшина Оксана Миколаївна, канд. пед. наук, доцент

Після аналізу звіту/звітів зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається до захисту.	відповідає
2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована.	
3	Виявлені запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Виявлені запозичення частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнуті. Робота може бути допущена до захисту після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укріття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені у роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системою перевірки на плагіат Anti-Plagiarism виявлено схожість з деякими документами у частині загальноживаних обов'язкових словосполучень у стандартних бланках, у структурі змісту, назвах назвах розділів/підрозділів, рамках форм, у назвах та URL-адресах публікацій переліку джерел посилання;

2) запозичення, виявлені у тексті роботи, є фрагментарними.

Максимальний обсяг запозичень, визначений системою Anti-Plagiarism, складає 1.0% з одного джерела. Загальна сумарна подібність у базі даних складає 5% за символами та 7% за лексемами.

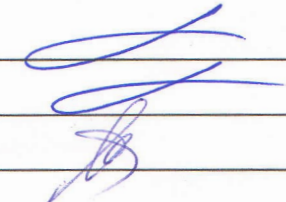
Крім того, за результатами додаткового аналізу коефіцієнт подібності 1 становить 4.95%, коефіцієнт подібності 2 – 0%. Виявлено 12 мікропробілів, 1 зайвий білий знак. Маніпуляцій з інтервалами не виявлено. З урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Дата 1.06.2020

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи



Леонід БЕДРАТЮК

Леонід БЕДРАТЮК

Оксана ЯШИНА