

Хмельницький національний університет  
Факультет програмування  
та комп'ютерних і телекомунікаційних систем  
Кафедра інженерії програмного забезпечення

## ДИПЛОМНИЙ ПРОЕКТ

Довідково-інформаційна система

Назва теми

«Ресторан»

Рівень вищої освіти Перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного  
забезпечення»

Шифр ДППЗ.170107.01.07.ПЗ

Виконав студент IV курсу група ПЗ-17-1

Керівник ст. викладач

Науковий ступінь, звання

Нормоконтролер канд. техн. наук, доцент

До захисту допускаю:

Завідувач кафедри інженерії  
програмного забезпечення

15 червня 2021 р.

Підпис

Я. І. Красюк

Ініціали, прізвище

Підпис

Г. І. Бедратюк

Ініціали, прізвище

Підпис

Г. І. Радельчук

Ініціали, прізвище

Підпис

Л. П. Бедратюк


Ініціали, прізвище

Хмельницький 2021

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Програмування та комп'ютерних і телекомунікаційних систем  
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 

Д. П. Бедратюк

05 02 2021 р.

## ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ)

Красюк Яні Іванівні

Прізвище, ім'я, по батькові студента

1. Тема проєкту (роботи) Довідково-інформаційна система «Ресторан»

Керівник проєкту (роботи) Бедратюк Ганна Іванівна, ст. викладач

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

Затверджена наказом ректора університету від 05.02.2021 р. № 11

2. Строк подання студентом проєкту (роботи) на кафедру 01.06.2021 р.

3. Вихідні дані до проєкту (роботи) Матеріали переддипломної практики

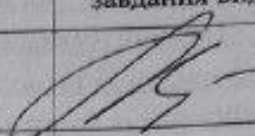
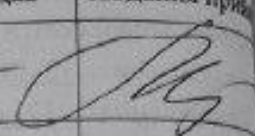
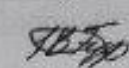
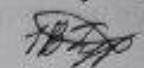
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Дослідження предметної області та постановка задачі, проєктування програмного забезпечення, програмна реалізація, тестування програмного забезпечення

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

Презентаційні матеріали (слайди)

6. Консультанти розділів дипломного проекту


Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Радельчук Г. І., доцент кафедри ІПЗ		
Антиплагіат	Гурман І. В., доцент кафедри ІПЗ		

7. Дата видачі завдання « 05 » лютого 2021р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітки
1 Ознайомлення з тематикою дипломного проектування (ДП), визначення та узгодження індивідуальних тем ДП	01.12 – 30.12.2020	
2 Дослідження предметної області, в якій планується використання програмного засобу (ПЗ), визначення задач та вимог, розробка технічного завдання	02.01 – 31.01.2021	
3 Проектування програмного забезпечення	01.02 – 28.02.2021	
4 Програмна реалізація	01.03 – 10.04.2021	
5 Тестування програмного забезпечення	11.04 – 30.04.2021	
6 Написання вступу, загальних висновків, оформлення джерел посилання та додатків. Оформлення пояснювальної записки ДП згідно вимог стандартів	01.05 – 25.05.2021	
7 Попередній захист ДП	Травень 2021 (згідно графіка)	
8 Перевірка ДП на плагіат, нормоконтроль, отримання відгуків та рецензій. Брошування (зшиття) пояснювальної записки	26.05 – 30.05.2021	
9 Підготовка до захисту та захист ДП	з 01.06.2021	

Студент

  
Підпис

Я. І. Красюк  
Ініціали, прізвище

Керівник проекту (роботи)

  
Підпис

Г. І. Бедратюк  
Ініціали, прізвище

## АНОТАЦІЯ

Тема дипломного проекту: Довідково-інформаційна система «Ресторан».

Автор проекту: Красюк Яна Іванівна.

Керівник проекту: Бедратюк Ганна Іванівна.

Пояснювальна записка: 69 с., 22 рис., 9 табл., 3 дод., 29 джерел.

Графічна частина: 19 слайдів.

ДОВІДКОВО-ІНФОРМАЦІЙНА СИСТЕМА, РЕСТОРАН, ВЕБ-ДОДАТОК,  
PHP, MY SQL.

Метою проекту є розробка довідково-інформаційної системи «Ресторан» для автоматизації роботи ресторану.

У дипломному проекті було проведено дослідження предметної області та виділення функціональних та нефункціональних вимог до розроблюваної довідково-інформаційної системи та здійснено постановку задачі. Було проведено аналіз існуючих довідково-інформаційних систем та дана їх характеристика, розроблено структуру довідково-інформаційної системи та спроектовано базу даних та інтерфейс користувача. Був наданий аналіз та проведено вибір технологій та методів реалізації програмного забезпечення. Проведена реалізація бази даних та модулів системи. Розроблена інструкція користувача.

1.06.2021

Дата



Підпис



## ЗМІСТ

Вступ.....	5
1 Дослідження предметної області та постановка задачі .....	7
1.1 Змістовний аналіз та опис предметної області, її структурних і функціональних особливостей .....	7
1.2 Аналіз наявного програмно-технічного забезпечення предметної області.....	17
1.3 Визначення вимог до програмного забезпечення та розробка технічного завдання.....	19
2 Проектування програмного забезпечення .....	21
2.1 Проектування архітектури програмного забезпечення.....	21
2.2 Детальне проектування програмного забезпечення.....	27
2.3 Проектування інтерфейсу користувача .....	29
3 Програмна реалізація .....	43
3.1 Структура і функціональне призначення модулів системи, їх взаємозв'язок. ....	43
3.2 Інструкція користувача.....	48
4 Тестування програмного забезпечення.....	54
4.1 Вибір та обґрунтування методів тестування системи .....	54
4.2 Розробка тестових наборів даних .....	57
Висновки .....	61
Перелік джерел посилання.....	62
Додаток А Технічне завдання.....	66
Додаток Б Програмний код .....	71
Додаток В Презентаційні матеріали.....	84

					ДПШЗ.170107.01.07.ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Виконав</i>		<i>Красюк Я.І.</i>			Довідково-інформаційна система «Ресторан». Пояснювальна записка	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Керівник</i>		<i>Бедратюк Г.І.</i>					4	98
<i>Н. Контр.</i>		<i>Радельчук Г.І.</i>			ХНУ, ПЗ-17-1			
<i>Зав. Каф.</i>		<i>Бедратюк Л.П.</i>						

## ВСТУП

Однією з перспективних в наш час галузей, що розвиваються, є ресторанний бізнес. Досягнення позитивних результатів у розвитку будь-якого бізнесу, а саме ресторанного, залежить від багатьох факторів. Процес керування ресторанним бізнесом завжди є доволі складним та вимагає: контролю за всім обліковим процесом та поведінкою персоналу закладу, аналізу транзакцій, обліку надходження продуктів, формування вартості страв та напівфабрикатів, процедур списання прострочених продуктів, дотримання санітарних і технологічних норм. Потреба у автоматизації всіх цих процесів постає з необхідності врахування великої кількості різних деталей. Зручність автоматизації процесів у всіх закладах громадського харчування досить очевидна не тільки з погляду власника, але і з позиції клієнтів, тому що довідково-інформаційні системи дають змогу персоналу закладу виконувати розрахунки з відвідувачами оперативніше, забезпечити дієву систему знижок для вже зареєстрованих клієнтів, здійснювати обслуговування клієнтів на основі черги, забезпечити дане меню усіма необхідними інгредієнтами.

В наш час актуальною стає реклама в системі Інтернет так як число користувачів в цій постійно збільшується. Якщо сайт правильно організований та привабливий для користувача то кількість відвідувачів ресторану може збільшитись в рази. Це досить недорогий і актуальний, в наш час, засіб для розкрутки та просування ресторану. Можливість оперативного розміщення нової інформації або швидке корегування існуючої. Реагування та зворотній зв'язок з клієнтами, що дозволяє проводити опитування на рахунок якості страв та обслуговування та їх побажань стосовно роботи ресторану. Потрібно наголосити, що довідково-інформаційна система це основний помічник в підтриманні позитивного іміджу закладу та його реформування відповідно до побажань клієнтів.

										Арк.
										5
Змн.	Арк.	№ докум.	Підпис	Дата	ДППЗ.170107.01.07.ПЗ					

Мета проекту – розробка довідково-інформаційної системи для автоматизації роботи ресторану.

Об’єкт дослідження – процес організації взаємодії відвідувачів ресторану з його працівниками за допомогою довідково-інформаційної системи.

Предмет – система створення довідково-інформаційної системи для відвідувачів ресторану.

Виходячи із мети, об’єкту та предмету дослідження можна окреслити такі завдання як:

- дослідити предметну область;
- здійснити аналіз існуючих програмних рішень даної галузі;
- розробити технічне завдання;
- спроектувати розробку програмного забезпечення;
- розробити інструкцію користувача.

					ДППЗ.170107.01.07.ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

# 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовний аналіз та опис предметної області, її структурних і функціональних особливостей

У ситуації, що склалася на ринку ресторанного бізнесу де конкуренція з кожним роком все зростає, та збільшується вимогливості клієнтів до якості послуг, що надаються, особливу увагу потрібно приділяти заходам по залученню і утриманню клієнтів підприємств громадського харчування. Просування послуг ресторану розглядаються як комплекс заходів по доведенню інформації до кінцевого споживача про ресторан та його переваги. Тому необхідно привернути увагу клієнта і переконати його скористатися запропонованими послугами.

На даний момент робота ресторану пов'язана з наявністю великої кількості інформації про меню, особисті данні співробітників, рахунки тощо. Крім того, підприємствам громадського харчування необхідно вирішувати такі важливі завдання, як збільшення доходу та управління змінами в ресторані, аж до зміни концептуальної спрямованості, наприклад, на підставі результатів опитування клієнтів ресторану.

Актуальність роботи визвана тим, що переважна більшість підприємств громадського харчування в якості інформаційної підтримки своєї діяльності використовує web-сайти, основним завданням яких є надання інформації про концепцію підприємства, меню, діючі спеціальні пропозиції та інше. Одночасно з цим адміністрація ресторану може впливати на ступінь лояльності клієнтів, впроваджуючи на сайті додаткові сервіси.

Звідси випливає, що наявність своєї візитної картки в мережі Інтернет дозволить забезпечити якісну інформаційну підтримку будь-якого підприємства громадського харчування.

					ДППЗ.170107.01.07.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Об'єктом роботи є ресторан, який в даний час не має свого сайту.

Предмет дослідження – інформаційні технологічні процеси, які використовуються в ресторанній діяльності.

Мета роботи – розробка сайту для ресторану «Домашній».

Практична значимість полягає в можливості використання спроектованого сайту для інформаційної підтримки діяльності ресторану «Домашній», в першу чергу, для завоювання лояльності відвідувачів ресторану.

Ресторанний бізнес – це сфера підприємницької діяльності, заснована на наданні послуг, метою яких є задоволення потреб в харчуванні на сервісній основі і як наслідок – отримання прибутку

Присутність активної конкурентної боротьби на ринку ресторанних послуг змушує менеджмент підприємств виконувати певні вимоги, наприклад, такі, як необхідність збільшення продуктивності і посилення конкурентної позиції, що неминуче пов'язане з впровадженням в повсякденне життя сучасних форм обслуговування. В основі таких форм – анімаційні програми, за допомогою яких збільшиться кількість відвідувачів в залі підприємства громадського харчування.

Громадське харчування – це сфера надання послуг, а значить головна задача даного сектора – зробити людей задоволеними на той час, що вони знаходяться в ресторані (кафе, барі або іншому типі підприємства).

Підприємництво в ресторані є успішним, якщо головною метою є задоволення потреб відвідувачів. Власник ресторану і колектив зацікавлені у формуванні сприятливих умов в своєму закладі для підвищення його популярності. Адже при великій конверсії відвідуваності значно збільшиться сума середнього чека на одного відвідувача, що, в свою чергу, вплине на збільшення обсягу продажів. Діяльність ресторану можна представити у вигляді системи, що складається з двох підрозділів: виробництво і збут продукції. Головні цілі цієї системи – задовольнити потреби конкретних споживачів і отримати прибуток на основі дослідження і прогнозування ринку. Для

									ДППЗ.170107.01.07.ПЗ	Арк.
										8
Змн.	Арк.	№ докум.	Підпис	Дата						

досягнення поставлених цілей вивчається внутрішнє і зовнішнє середовище підприємства, розробляється стратегія і тактика поведінки на ринку за допомогою маркетингових технологій.

В основі даних технологій лежить певний комплекс заходів щодо поліпшення якості продукції, по розширенню асортименту послуг, з вивчення споживачів і конкурентів, по забезпеченню цінової політики, щодо формування, оптимізації, реклами, каналів руху товару і організації збуту.

Виробнича діяльність з продажу ресторану починається з сегментування ринку, тобто з розбивки його на певні групи споживачів, для кожної з яких повинні бути запропоновані різні види продукції і послуг. Від того, наскільки правильно обрані сегменти ринку, багато в чому залежить успіх ресторану в конкурентній боротьбі.

Заклад сфери ресторанного бізнесу повинен вирішити, скільки сегментів ринку вони в змозі охопити, і зосередитися на самих вигідних.

На сьогоднішній день в індустрії громадського харчування в Україні є велика кількість підприємств з різним рівнем сервісу, якістю продукції і різноманітністю використовуваного обладнання.

До 30-х років ХХ століття галузь не отримувала такого розвитку, так як увага всієї країни була спрямована на загальну індустріалізацію та організацію колгоспного господарства. У повоєнні роки мережа підприємств громадського харчування почала стрімко розширюватися: збільшився асортимент страв, покращилося забезпечення підприємств технологічним та холодильним устаткуванням, підвищилася культура під час обслуговування відвідувачів. Певною особливістю радянської системи громадського харчування було одномоментне обслуговування значної кількості населення в промисловості і сільському господарстві. Негативним моментом даної системи була вимушена обмеженість в асортименті. У цей період підприємства громадського харчування активно займалися виїзною реалізацією готової продукції, що також створювало певні труднощі. В кінці 80-х років ХХ століття багато підприємств

										ДППЗ.170107.01.07.ПЗ	Арк.
											9
Змн.	Арк.	№ докум.	Підпис	Дата							

громадського харчування почали працювати тільки на основі господарського розрахунку, тобто вони мали самостійний баланс. Почали відкривати кооперативні підприємства громадського харчування, хоча велика частина з них ще залишалася в складі державних торговельних організацій.

В період переходу на ринкові відносини на початку 90-х років ХХ століття стали формуватися невеликі, самоокупні компанії з високим рівнем сервісу і якістю приготування їжі. Характерною особливістю таких підприємств громадського харчування є поєднання в ньому функцій виробництва, а також головним є споживання.

Якщо переважна більшість підприємств суміжних галузей обмежували себе виконанням лише однієї або двох функцій, то підприємства громадського харчування могли виконувати три взаємопов'язані між собою функції такі як: виробництво кулінарної продукції, продаж продукції, організація її споживання.

Вихідною вважається функція виробництва, витрати праці якої складають 70-90% всіх витрат в галузі. В процесі виробництва створюється новий продукт. Продукція власного виробництва надходить в реалізацію до нового споживача з якостями, характеристиками і збільшеною вартістю. Харчування – це форма споживання. В основі різноманіття підприємств громадського харчування лежить їх класифікація, в основу якої закладено такі ознаки: асортимент реалізованого товару і види послуг для населення, обсяг і різновид послуг, ступінь зручності і якості обслуговування, форми організації виробництва, циклічність функціонування протягом 12 місяців, рівень мобільності (місце дислокації), спосіб розміщення підприємств в будівлях.

Найчастіше підприємства громадського харчування розрізняються за характером виробництва, та асортиментом продукції, що випускається, по обсягу і виду послуг, що надаються. Найбільш поширеними підприємствами громадського харчування вважаються ресторани, кафе, бари, їдальні, заклади швидкого обслуговування. Вони можуть працювати на сировині чи напівфабрикатах, бути в системі мережі або незалежними, з різною формою

										Арк.
										10
Змн.	Арк.	№ докум.	Підпис	Дата	ДППЗ.170107.01.07.ПЗ					

власності. Підприємствам такого типу пред'являють досить жорсткі вимоги. Вони відносяться до зовнішнього вигляду підприємства та оформлення залів і приміщень для споживачів, наявності сцени і танцювального майданчика, банкетного залу або закритих (окремих) залів, мікроклімату, меблів, столового посуду і приладів, столової білизни, меню і асортименту своєї продукції і покупних товарів, методів обслуговування споживачів, одягу і взуття обслуговуючого персоналу, музичного супроводу.

В нашій і зарубіжній літературі можна наштотувати на найрізноманітніші думки про роль web-сайту в діяльності компанії. Будь-яке сучасне комерційне підприємство, що прагне до досягнення отримання максимального прибутку, повинно мати власний web-сайт, який є ефективним інструментом не тільки для збільшення обсягів продаж, але і з метою маркетингових досліджень в споживчому середовищі, а також організації зворотного зв'язку з покупцями. Це нематеріальний актив компанії, який відрізняє сучасний бізнес.

Компанія без сайту – це те ж саме, що ділова людина без візитної картки. Щоб користуватися можливостями web-сайту з максимальною ефективністю, необхідно чітко розуміти, які завдання вирішуються за допомогою сайту. В основі web-сайту повинна лежати певна бізнес-модель, яка є моделлю підтримки існуючого бізнесу, моделлю створення нового сучасного бізнесу в мережі Інтернет.

Web-сайт можна порівняти з традиційними виставками. Його можна уявити, як досить великий міжнародний виставковий зал, де у потенційних покупців є шанс вступити в безпосередній контакт з потенційними продавцями і відвідати презентовану онлайн експозицію. Сайт виконує роль не тільки рекламного інструменту, але і інструменту особистих продажів. Крім цього, web-сайт - це щось середнє між прямими продажами (підтримання діалогу зі споживачем) і рекламою (підвищення інформованості, демонстрація продукту і надання інформації без інтерактивної участі).

									Арк.
									11
Змн.	Арк.	№ докум.	Підпис	Дата					

ДППЗ.170107.01.07.ПЗ

Таким чином, рекламні комунікації, які створюють потік відвідувачів на сайті, повинні не тільки містити в собі активні і пасивні комунікації, а й залучати онлайн і офлайн-інструменти.

Адже при наявності власного сайту зростає ефективність проведених в мережі рекламних заходів. Інформація про відгуки відвідувачів на сайті може відразу потрапляти в інформаційну систему компанії і служити для більш якісного їх обслуговування. Варто зробити висновок, що сайт є основою системи комунікацій в Інтернеті.

В цілому перераховані маркетингові завдання, що виконуються сайтом, зводяться до наступних завдань: здійснення і підвищення рівня продаж, формування та підтримку іміджу, надання різного роду інформації, формування каналу зворотного зв'язку зі споживачем

Сайт компанії – один з головних комунікативних рекламних інструментів в мережі Інтернет. Споживачі компанії є головною цільовою аудиторією сайту, але не єдиною. За допомогою сайту компанія має можливість взаємодіяти з партнерами, акціонерами, ЗМІ та іншими контактними аудиторіями.

Офіційний сайт являє собою потужний механізм розвитку і підтримки іміджу компанії і її товарів, сприяє зростанню популярності бренду. Сайт повинен містити в собі всю потрібну інформацію про компанію, про сферу її діяльності, вироблені товари і послуги, що надаються. Представляючи собою найкращий маркетинговий майданчик, грамотно розроблений корпоративний сайт неодмінно буде працювати над створенням у споживачів і партнерів образу перевіреної компанії, що йде в ногу з часом та динамічно розвивається.

Сайт може стати основним майданчиком для забезпечення зворотнього зв'язку з споживачами компанії.

Головні характеристики сайту можна охарактеризувати моделлю ,яка містить такі елементи:

1) Контекст (Context): баланс між естетичним і функціональним змістом сайту, його оформленням і дизайном. Сайт може бути:

										ДППЗ.170107.01.07.ПЗ	Арк.
											12
Змн.	Арк.	№ докум.	Підпис	Дата							

– функціонально орієтованим, якщо основну роль грають функції, на виконання яких він спрямований;

- естетично орієтованим, коли його вигляд web-сторінки є в пріоритеті;
- гібридним, якщо корпоративний сайт ефективний естетично.

2) Зміст (Content): тексти, фотографії, зображення, музика і відео.

3) Ступінь об'єднання (Community): способи, які дозволяють всім користувачам сайту підтримувати зв'язок один з одним. за здатністю корпоративних сайтів забезпечити підтримку. У цьому випадку їх можна умовно розділити на три види:

- сайти, які не дають користувачам можливість взаємодіяти один з одним.
- обмежені сайти, які пропонують такі можливості, як читання і розміщення відгуків про товари і послуги компанії;
- сайти, які пропонують інтерактивні функції, як чати і дошки оголошень.

4) Комунікації (Communication): здатність сайту здійснювати двосторонній зв'язок між представником компанії і споживачем.

5) Зв'язок (Connection): ступінь взаємозв'язку конкретного сайту з іншими сайтами за допомогою розміщення посилань, партнерських програм.

6) Комерція (Commerce): можливості корпоративних сайтів по здійсненню комерційних угод. Так, сайти компаній, які здійснюють електронну торгівлю, мають можливість зареєструвати клієнтів, зберігати інформацію про переваги, контакти замовника і параметри доставки зробленого замовлення.

7) Кастомізація (Customization): здатність сайту надавати інформацію та додаткові можливості різних сегментів споживачів, можливість персоналізації

В якості ключових компонентів корпоративного сайту виділяють:

- наявність контактних даних (адреса, телефон та ін.);
- зрозумілий інтерфейс (логічна структура, читабельність текстів);
- зовнішня привабливість;
- наявність корпоративної інформації (звіти, посередники і точки продажів, можливості працевлаштування);

									ДППЗ.170107.01.07.ПЗ	Арк.
										13
Змн.	Арк.	№ докум.	Підпис	Дата						

- можливості персоналізації і збереження персональних налаштувань;
- відстеження знаходження товару по трек-номеру;
- можливості здійснення онлайн-покупки (можливість оплати, особливості ціноутворення).

Сайт повинен включати таку характеристику, як демонстрація надійності, і бути заснованим на корпоративному іміджі та репутації, які існують за межами мережі Інтернет [1] .

Крім цієї характеристики до головної складової сайту відносять:

- тип доменного імені;
- інформацію про продукт;
- мовні можливості ;
- можливості персоналізації;
- контактна інформація.

Тому, існує перелік характеристик web-сайту комерційної компанії.

Основні характеристики сайту комерційної компанії представлені в характеристиках таблиці 1.1.

Таблиця 1.1 Основні виділені характеристики сайту

Ч.ч.	Характеристика
1	2
Інформація	
1	Наявність інформації про функції, властивості, ціни товару
2	Наявність корпоративної інформації (фінансові звіти, вакансії, мета)
3	Наявність контактної інформації (телефон, адреса)
4	Наявність інформації про тенденції галузі, галузеві огляди
5	Наявність розділу «Питання які найчасіше задають»
6	Наявність відгуків про компанію і товари
7	Історія компанії

Кінець таблиці 1.1

1	2
Привабливість	
8	Наявність фото товару та відео про товар
9	Оформлення і привабливість сайту
10	Фотографії клієнтів
11	Фотографії персоналу компанії та її керівника
Простота	
12	Простота навігації
13	Ясність та зрозумілість тексту
Додаткові можливості	
14	Наявність інструкцій по використанню, поради по підборі моделей, кольору, розміру
15	Можливість збереження персональних налаштувань
16	Наявність реклами інших компаній
17	Наявність онлайн гри та інших розваг
18	Можливість безкоштовного скачування додаткових програм
Зворотній зв'язок	
19	Можливість здійснити онлайн-покупку
20	Онлайн-діагностика виникаючих проблем при використанні товару
21.	Можливість відстежувати статус замовлення і його місце знаходження

Компанія, де цільова аудиторія – споживачів, які отримують фірмовий товар і як правило, не потребують додаткових вкладень коштів в розвиток. Основним завданням такої компанії буде використання свого web-сайту як додаткового інструменту формування іміджу. Головними вимогами до подібних сайтів зазвичай є: зовнішня привабливість і дизайн, інформація про товар і та спеціальні пропозиції.



5) Інші види контенту. Нинішні технології пропонують досить багато цікавих рішень для створення і доставки контенту, включаючи відеотрансляції в реальному часі (прямі ефіри), інтерактивні карти та інше. Перелік видів контенту буде рости з появою нових технологій і потреб, і цей список можна буде весь час доповнювати.

## 1.2 Аналіз наявного програмно-технічного забезпечення предметної області

На ринку програмного забезпечення існує безліч довідково-інформаційних систем які представляють різні види ресторанів. Розглянемо деякі з них.

Цукерня Потоцьких. Дана довідково-інформаційна система, на мою думку, не дає достатньої інформації про заклад. Вона більше пристосована під соціальну мережу Instagram чи Facebook де в основу покладено обмін фотографіями та відео.

Відсутнє меню та опис страв. Невідомо чи можна зробити замовлення. Хоча сайт виглядає доволі привабливо але не для всіх зрозуміла яка головна концепція цього закладу.

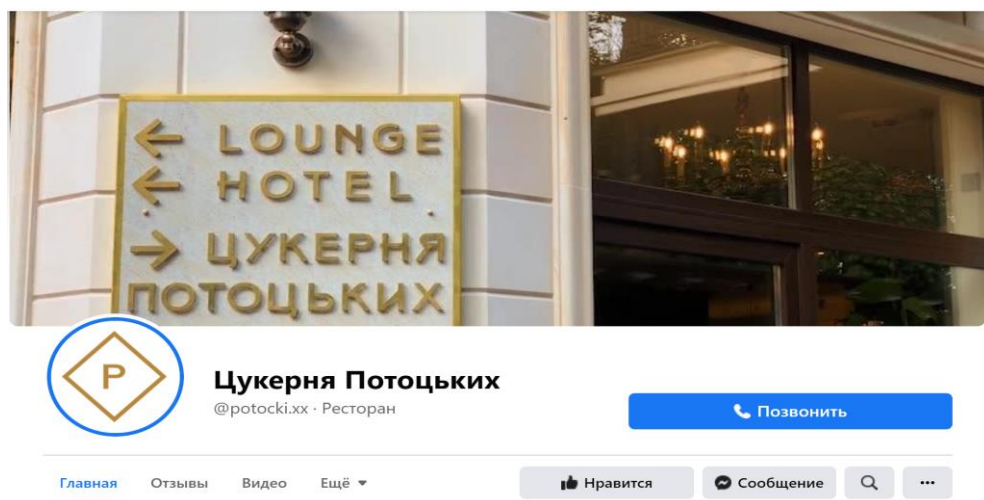


Рисунок 1.1 - Фрагмент сайту ресторану «Цукерня Потоцьких»

									Арк.
									17
Змн.	Арк.	№ докум.	Підпис	Дата					

ДППЗ.170107.01.07.ПЗ



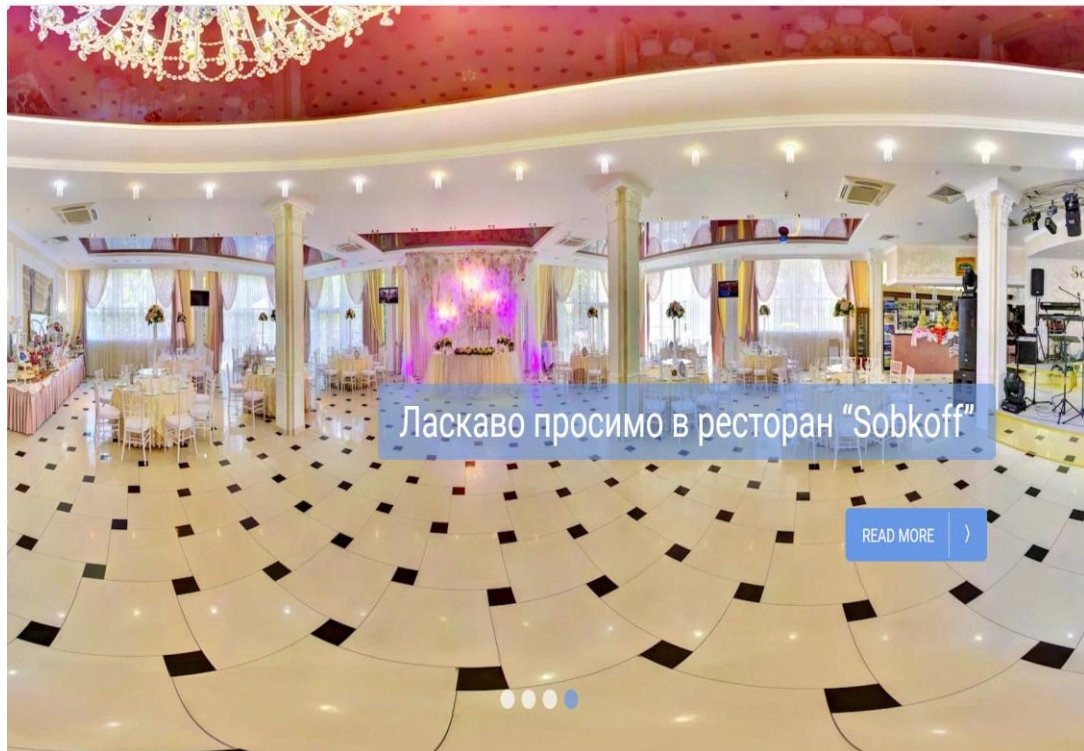


Рисунок 1.3 – Фрагмент сайту ресторану «Sobkoff»

Як результат приваблива довідково-інформаційна система з зручним інтерфейсом, зрозумілою і привабливою навігацією, можливістю зворотного зв'язку. Але не з всією довідковою інформацією про даний заклад, що може відобразитись на відвідуванні.

### 1.3 Визначення вимог до програмного забезпечення та розробка технічного завдання

Основною метою даного дипломного проекту є створення довідково-інформаційної системи «Ресторан». Ця система повинна бути забезпечена зручним та зрозумілим інтерфейсом, в якому буде розглянуто такі розділи як:

- меню;

									Арк.
									19
Змн.	Арк.	№ докум.	Підпис	Дата	ДПШЗ.170107.01.07.ПЗ				

- контактна інформація;
- відгуки;
- страва дня;
- опис блюда з фото;
- можливість замовити он-лайн з доставкою;
- здійснення оплати платіжною картою;
- зміна мови;
- години роботи.

На основі проведеного аналізу вимог було розроблено технічне завдання, яке знаходиться у додатку А.

У даному розділі було проведено дослідження предметної області та виділення функціональних та нефункціональних вимог до розроблюваної довідково-інформаційної системи та здійснено постановку задачі. Було проведено аналіз існуючих довідково-інформаційних систем у мережі Інтернет та дана їх характеристика.

					ДППЗ.170107.01.07.ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Проектування архітектури програмного забезпечення

Архітектура довідково-інформаційної системи визначає концепцію, що створює структуру і взаємозв'язки між компонентами інформаційної системи.

Найбільш вживаними та вдалимими в наш час вважаються інформаційні довідкові системи з використанням CMS . Тому для реалізації було вибрано саме її.

Дана довідково-інформаційна система містить такі компоненти як плагіни, шаблони, теми, сторінки та базу даних. Саму архітектуру довідково-інформаційної системи можна розглянути на рис.

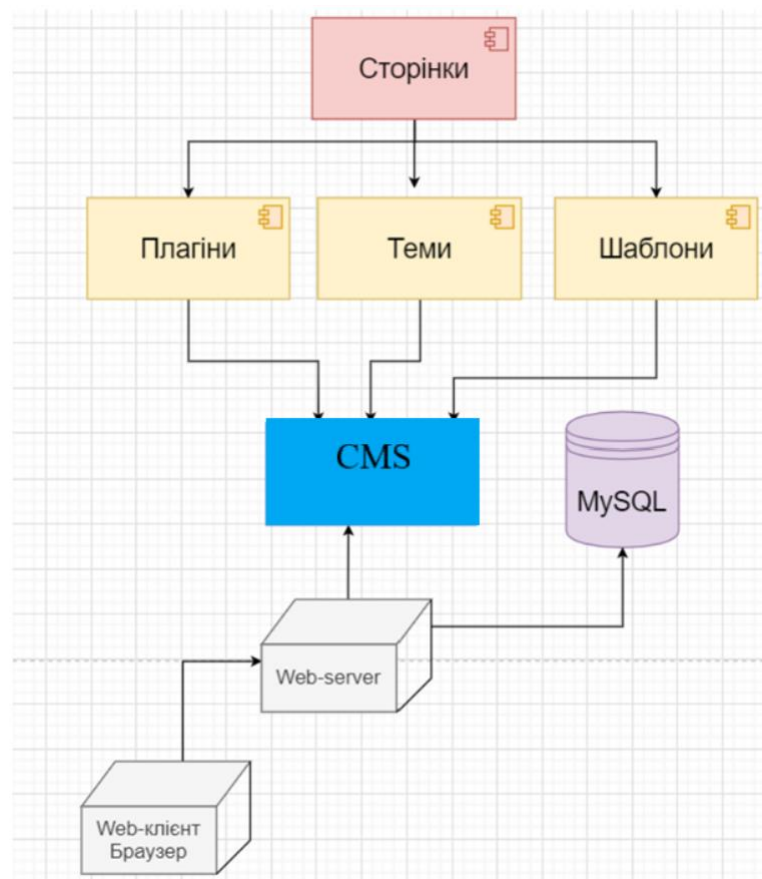


Рисунок 2.1 – Архітектура інформаційної системи

Змн.	Арк.	№ докум.	Підпис	Дата

Шаблони та теми включають в себе набір файлів, які відповідають за зовнішній вигляд проекту який розробляється. Ці дані зберігаються в папці wp-content/themes.

Плагіни є доповненнями, які розширюють функціональність і допомагають у розробці проекту . Вони зберігаються у папці wp-content/plugins.

В базі даних яка створена зберігається вся інформація, що представлена в довідково-інформаційній системі. Тут є інформація про користувачів, різні налаштування, інформація про послуги, меню, складові блюд [2].

Сторінки становлять статичні елементи на яких розміщується вся інформація яка призначена для відображення користувачам. Відображення вмісту файлів проекту зображено на рисунку 2.2

Имя	Дата изменения	Тип	Размер
wp-admin	07.05.2020 17:58	Папка с файлами	
wp-content	28.05.2020 11:43	Папка с файлами	
wp-includes	07.05.2020 17:58	Папка с файлами	
.htaccess	07.05.2020 18:25	Файл "HTACCESS"	1 КБ
index.php	06.02.2020 08:33	Файл "PHP"	1 КБ
license.txt	29.04.2020 23:19	Текстовый докум...	20 КБ
readme.html	29.04.2020 23:19	Chrome HTML Do...	8 КБ
wp-activate.php	06.02.2020 08:33	Файл "PHP"	7 КБ
wp-blog-header.php	06.02.2020 08:33	Файл "PHP"	1 КБ
wp-comments-post.php	06.02.2020 08:33	Файл "PHP"	3 КБ
wp-config.php	15.05.2020 11:11	Файл "PHP"	4 КБ
wp-config-sample.php	01.04.2020 00:06	Файл "PHP"	3 КБ
wp-cron.php	06.02.2020 08:33	Файл "PHP"	4 КБ
wp-links-opml.php	06.02.2020 08:33	Файл "PHP"	3 КБ
wp-load.php	06.02.2020 08:33	Файл "PHP"	4 КБ
wp-login.php	10.02.2020 05:50	Файл "PHP"	47 КБ
wp-mail.php	29.04.2020 23:19	Файл "PHP"	9 КБ
wp-settings.php	29.04.2020 23:19	Файл "PHP"	19 КБ
wp-signup.php	06.02.2020 08:33	Файл "PHP"	31 КБ
wp-trackback.php	06.02.2020 08:33	Файл "PHP"	5 КБ
xmlrpc.php	06.02.2020 08:33	Файл "PHP"	4 КБ

Рисунок 2.2 – Вміст файлів проекту

Пройшовши етап дослідження всіх складових, що мають відношення до проекту було визначено ціль його проведення та задачі які необхідно виконати. Для цього було сформовано перелік вибраних функцій які будуть включені в майбутній додаток, та способи їх втілення.

Наступний етап – це етап створення веб-додатку, а це говорить про те що буде використано розробку новітні ІТ- технологій, які ще не відомі широкому загалу, але необхідні для задоволення потреби користувачів.

Початком розробки проекту можна вважати створення контекстної діаграми. Вона містить короткі, місткі, лаконічні дані які показують особливості взаємодії довідково-інформаційної системи та оточення і описують їх зміст.

Можна визначити одні із важливих складових цієї діаграми:

- інформація та матеріали, які є в частині ідентифікації продукту та являються даними вхідного характеру;
- результати до яких призведе втілення продукту в життя та які є даними вихідного характеру;
- дані інформаційного та матеріального характеру, які допомагають реалізації проекту та є даними з менеджменту;
- дані які є інструментами це – персонал, технічне забезпечення, програми які проводять аналіз важливих елементів діаграми;
- замовлення клієнта на веб-додатку це є вхідні дані;
- вихідні дані- замовлення яке вже готове та видача чеку;
- правила сервісу, закон України, умови оплати для клієнта відносяться до управління;
- механізми – це адміністратор та ресторатор.

Діаграма має такий вигляд як зображено на рисунку 2.3

					ДППЗ.170107.01.07.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

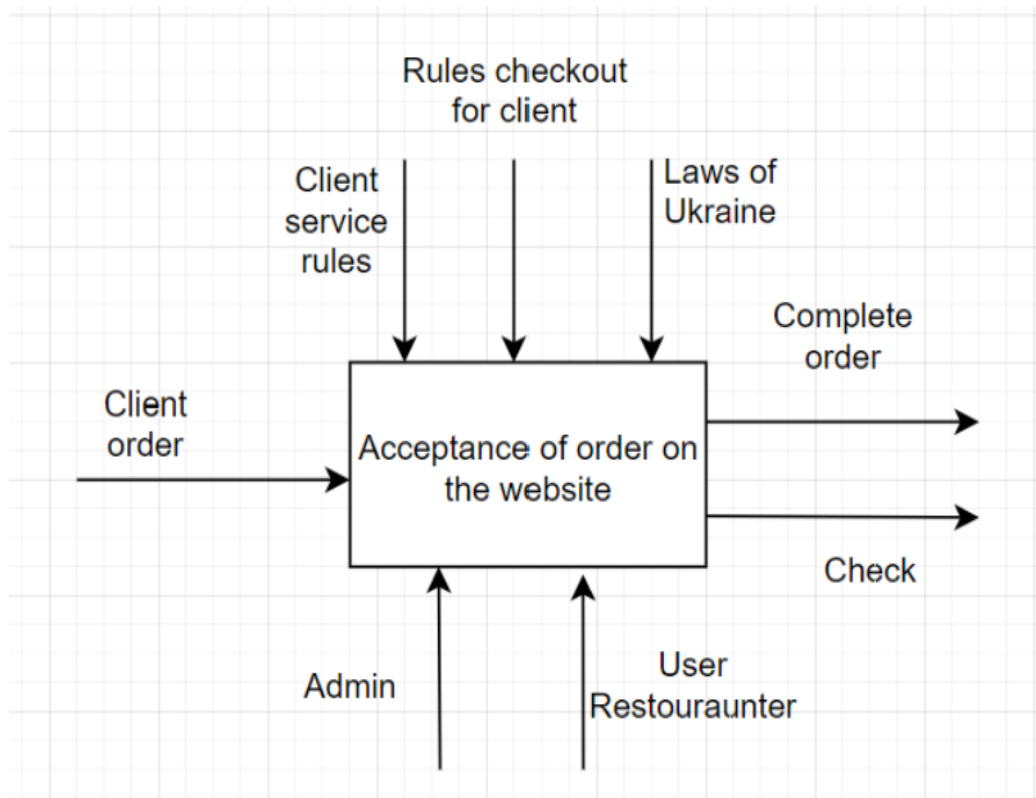


Рисунок 2.3 – Контекстна діаграма розробки веб-додатку

Основна властивість контекстної діаграми полягає в зображенні основної суті веб-додатку, що потребує необхідності створення декомпозиції. Декомпозиція надає дані про особливості роботи веб-додатку.

Принципи декомпозиції:

- клієнтське замовлення;
- обрання страви з меню;
- звіт по оплаті - зроблена оплата чи ні.

Якщо оплата відбулася – то супроводжується видача чеку, якщо ні то звернення до служби підтримки.

В діаграмі варіантів використання важливою частиною є планування проекту. Ця діаграма є найпростішою з поведінкових діаграм та демонструє результат взаємодії між акторами та прецедентами і дозволяє отримати інформацію про функціонал системи та показує як актором буде взаємодіяти у цій системі [3].

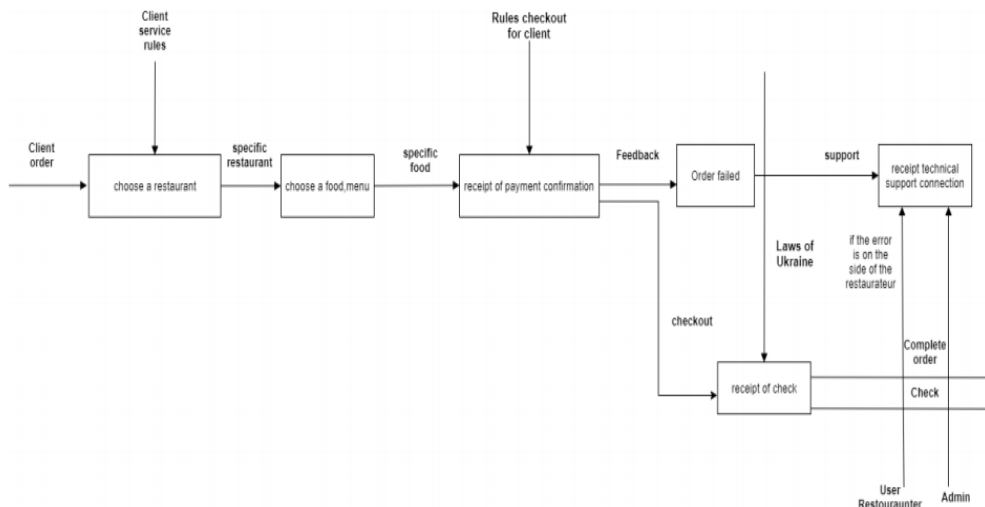


Рисунок 2.4 – Декомпозиція діаграми

В створеній діаграмі є такі актори як User Customer – це актори, які обирають певну систему та збираються взаємодіяти з нею. Після відбору всіх потенційних користувачів системи формується перелік всіх варіантів взаємодії з нею. Відповідно до вимог вибираються варіанти, яким відповідає веб-ресурс.

Діаграма Use Case розроблена завдяки інформації про користувачів веб-додатку та можливих варіантів взаємодії з ними.

Визначено три типи користувачів (юзерів).

Фіолетовим кольором визначається загальні дії. Решта відповідно до кольору кожного юзера.

Варіанти використання веб-додатку:

- реєстрація;
- авторизація;
- перегляд ресторанів та меню;
- перегляд історії оплати та заказів;
- залишати відгуки;

- розміщення ресторану та меню;
- перегляд заробітку та витрат ресторану;
- пошук по сайту;
- оплата онлайн;
- додавати в улюблене;
- підтримка юзерів;
- підтримка сайту;
- налаштування профілю в особистому кабінеті.

На основі зібраних даних про акторів та всіх варіантів використання веб-додатку розроблена Use Case діаграма.



Рисунок 2.5 – Діаграма варіантів використання

Змн.	Арк.	№ докум.	Підпис	Дата

## 2.2 Детальне проектування програмного забезпечення

Для кращого розуміння того як буде виглядати наш веб-додаток розроблено його архітектуру. Спроектовано логіку та розуміння як працює веб-додаток і що використано для його роботи.

Головним елементом веб-додатку є сервер, за допомогою якого функціонує весь додаток та база даних за допомогою якої проходить обмін даними. Компоненти, плагіни та модулі це ті складові, які приймають участь у створенні сайту

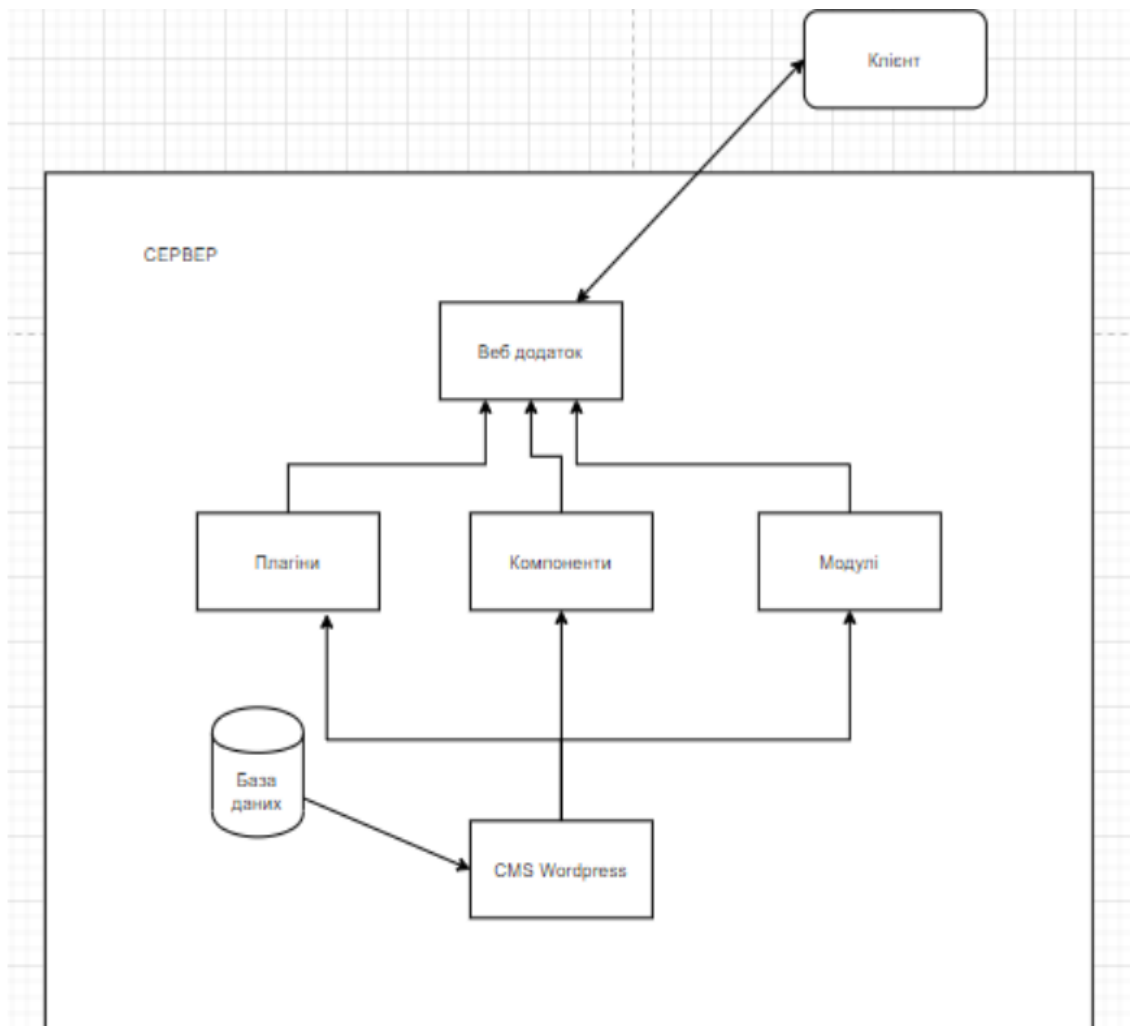


Рисунок 2.6 – Архітектура веб-додатку

Змн.	Арк.	№ докум.	Підпис	Дата

## 2.3 Проектування логічної моделі бази даних

Основою будь-якої інформаційної системи є база даних. Моделлю бази даних називається графічне або текстове представлення бази даних, де виділяються сутності, атрибути та зв'язки між ними.

Відповідно до побудованої моделі варіантів використання наша інформаційна система має зберігати всю інформацію в базі даних.

Наша база даних складається з 11 таблиць які необхідні для роботи програмного продукту.

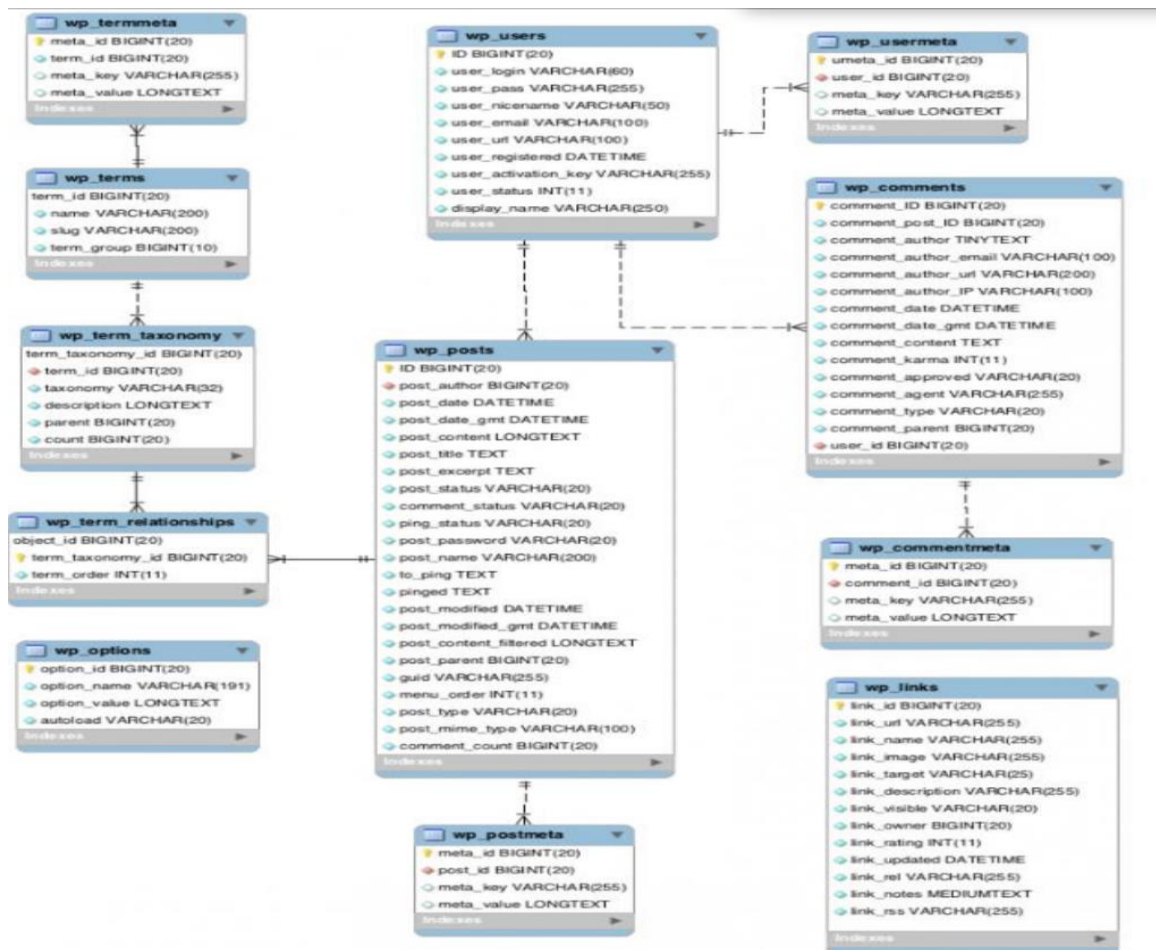


Рисунок 2.7 – База даних довідково-інформаційної системи

Змн.	Арк.	№ докум.	Підпис	Дата



– інтерфейс – одна з самих важливих частин довідково-інформаційної системи. Це якщо його розглядати з точки зору реклами та з метою продажу і з погляду безпосереднього користувача цієї системи, який може працювати з нею по декілька годин в день поспіль;

– інтерфейс впливає на характер рішень, що приймає потенційний відвідувач ресторану. Він може прискорювати для себе час прийняття рішення та покращувати або погіршувати цю якість;

Існують такі властивості, яким повинен задовольняти інтерфейс:

1) Адаптованість, це означає, що даний інтерфейс повинен бути: сумісним з багатьма потребами та можливостями даного користувача та забезпечувати йому простоту переходу від виконання однієї функції до іншої; мати змогу надавати користувачу вказівки відповідно його можливих дій, а також створювати відповідний зворотний зв'язок на його запити; та надати можливість цьому користувачу відчувати себе повноцінним керівником ситуації при розв'язанні всіх існуючих типів задач, тобто, забезпечувати його всією потрібною інформацією. Користувач має бути впевненим, що це він сам розв'язує дану поставлену задачу. Інтерфейс має забезпечувати користувача різноманітними, взаємно доповнюючими формами представлення результатів у відповідності від типу запиту або від його характеру отриманого рішення і враховувати всі особливості користувачів різних рівнів. Наприклад, для багатьох керівників підприємств кращим є узагальнене графічне представлення результатів діяльності довідково-інформаційної системи на прикладі діаграм та графіків, а інженеру-економісту потрібні дані з конкретними цифрами.

2) Достатність інтерфейсу означає, що всі допустимі запити користувача повинні бути зрозумілими і однозначними для користувачів всіх рівнів, та для прикладних задач всіх типів, а реакція даної системи на всі типи запитів - однозначною і зрозумілою і, по можливості, простою.

3) Дружність інтерфейсу – це найбільш простота у використанні та готовність максимально задовольнити запити користувача в процесі розв'язання

									Арк.
									30
Змн.	Арк.	№ докум.	Підпис	Дата					

визначеного класу задач.

4) Гнучкість інтерфейсу – це можливість його пристосування до розв’язання конкретної задачі. Якщо дана розв’язувана задача дуже складна, то даний інтерфейс повинен максимально полегшувати формулювання запитів та надавати результати у формі, що легко і швидко сприймається користувачем. Тобто інтерфейс має буди максимально простим навіть тоді, коли розв’язується дуже складна задача. При цьому простота це:

– інтерфейс не має бути перевантажений деталями коли подається представлення розв’язку поставленої задачі

– користувач може не завжди охопити всіх подробиць – тобто не повинно бути нічого зайвого;

– він не повинен мати зайвих декоративних деталей, що відволікають від основної задачі;

– інтерфейс має бути консистентним, тобто він має, ґрунтуватись на використанні знайомих, загальноприйнятих методів та засобів представлення даної інформації;

– в ідеалі, весь процес взаємодії користувача з системою не має представляти ніяких труднощів.

– Використовуючи всі ці характеристики було сформовано інтерфейс даної довідково-інформаційної системи ресторану.

– Головна сторінка якого містить основну інформацію про сам заклад. Його приблизне меню, де знаходиться сам ресторан та які страви у ньому є. Які обмеження існують під час пандемії та які нові послуги надає ресторан.

– Відвідувач може бачити страви дня, та склад блюд, що знаходиться зразу під фото, щоб відвідувач мґ оцінити і склад і його зовнішній вигляд [4 ].

										ДППЗ.170107.01.07.ПЗ	Арк.
											31
Змн.	Арк.	№ докум.	Підпис	Дата							

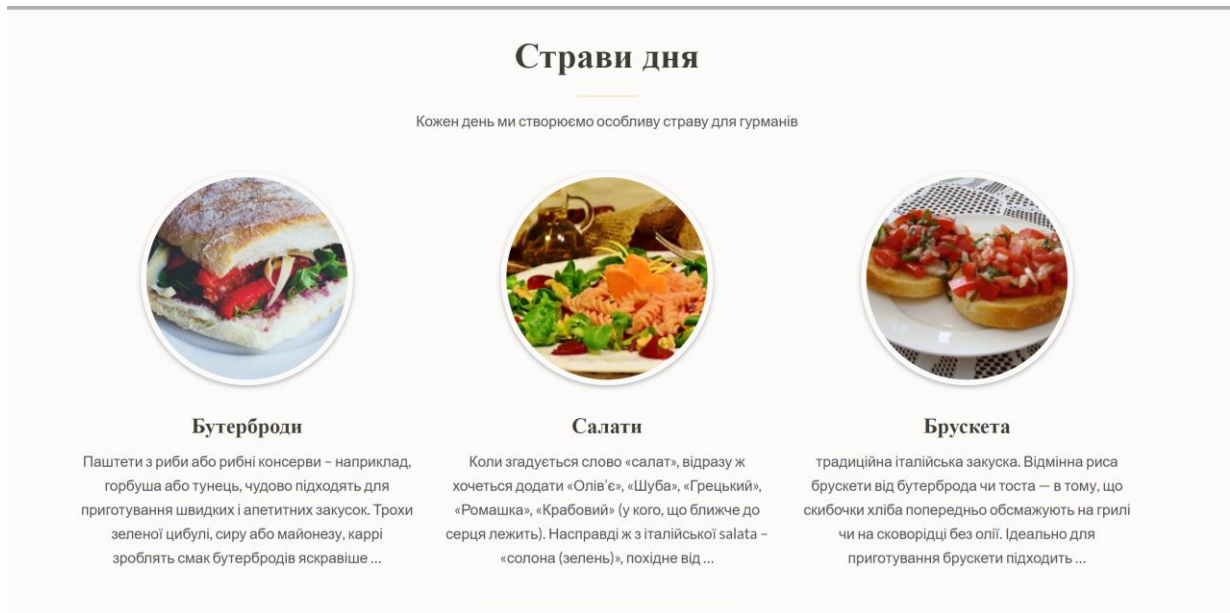


Рисунок 2.8 – Частина головної сторінки

Меню ресторану розбито на категорії по типам страв. Є можливість зареєструватися, бачити свої замовлення, оплату та вибрати тип розрахунку. Досить важливим є те, що можна зразу переглянути відгуки відвідувачів.

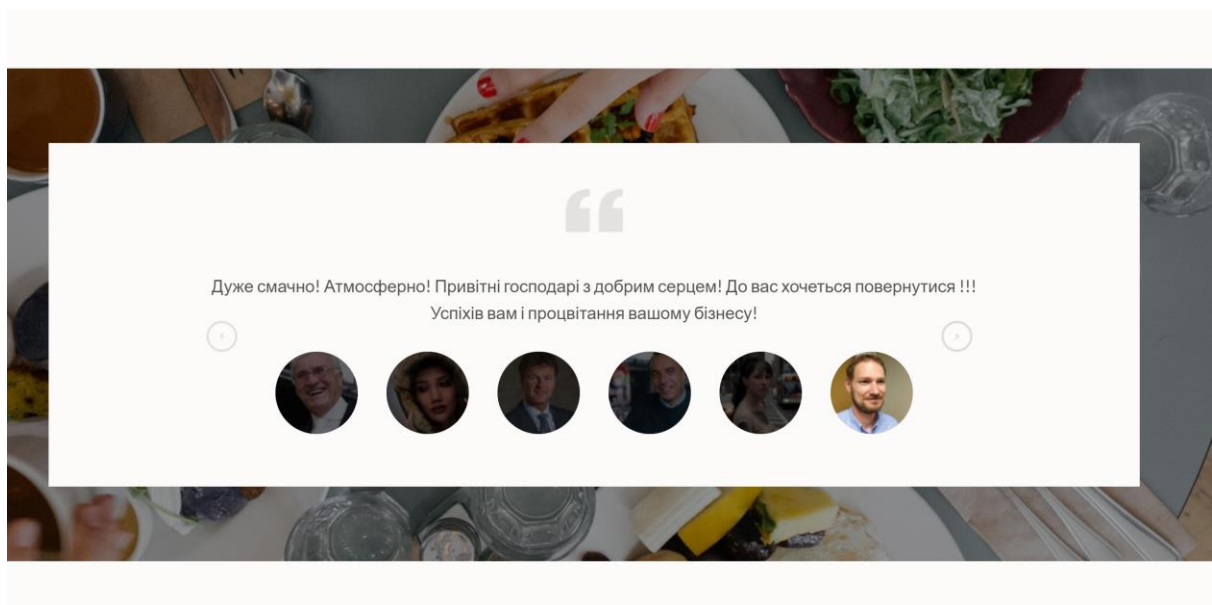


Рисунок 2.9 – Відгуки відвідувачів

Інтерфейс зручний, доступний, дружній до користувача. Всі потрібні дані знаходяться в легкому доступі для користувача.

Кольорова гама спокійна та сприяє позитивним емоціям, а головне не

										Арк.
										32
Змн.	Арк.	№ докум.	Підпис	Дата						

ДПШЗ.170107.01.07.ПЗ

відволікає від інформаційної складової нашої довідково-інформаційної системи.

### 2.3 Аналіз та вибір технологій і методів реалізації системи

Розвиток мережі Інтернет призвів до появи нової категорії програм – Web-додатків. До Web-додатків відносять набір Web-сторінок, сценаріїв і інших програмних засобів, розташованих на одному або декількох комп'ютерах (клієнтських і серверних) і об'єднаних для виконання прикладної задачі. При цьому Web-додатки, що публікують бази даних в Інтернет, являють собою окремий клас. Сучасні інформаційні системи, побудовані на основі Web-додатків, що використовують бази даних, базуються на багаторівневої клієнт-серверній архітектурі і принципах функціонування Інтернету. Web-додатки являють собою особливий тип програм, побудованих за архітектурою «клієнт-сервер». Особливість їх полягає в тому, що саме Web-додаток знаходиться і виконується на сервері, а клієнт при цьому отримує тільки результати роботи. Робота програми ґрунтується на отриманні запитів від користувача (клієнта), їх обробці та видачі результату. Передача запитів і результатів їх обробки відбувається через мережу Інтернет [5].

Сучасні Web-додатки стають все більш складними зі складною логікою. Раніше продуктивність таких додатків визначалася, в основному, швидкістю роботи використовуваного SQL-сервера і тим, чи знайдеться для нього досить дієва реалізація драйвера доступу до SQL-сервера для використаної мови програмування. Це виходить з того, що перше покоління Web-додатків буквально зчитувало і записувало інформацію в бази даних. Користувачів в даний момент було відносно небагато. Тому, час на відпрацювання SQL-запитів становив 70-90% від всього часу обробки HTTP-запиту. Із збільшенням кількості користувачів та нарощуванням логіки додатка вимоги до відповідної мови програмування і середовища виконання досить сильно зростають. До цього потрібно додати, що зовсім недавно Web-додатки перейшли зі світу

									ДППЗ.170107.01.07.ПЗ	Арк.
										33
Змн.	Арк.	№ докум.	Підпис	Дата						

Інтернет в площину корпоративних додатків. Це знову підняло вимоги до ефективності середовища виконання. У наш час для створення Web-додатків є безліч різних мов програмування. Найвідоміші з них – PHP, Perl, Java.

В даний час існують і успішно застосовуються різні види технологій побудови Web-додатків серверної сторони. Всі такі додатки мають загальну мету – реалізацію бізнес-логіки із сторони сервера та створення коду для клієнта. Додатково у всіх цих додатків однакова архітектура взаємодії сервера та клієнта і загальний протокол взаємодії – HTTP [8].

Для розробки web-додатків для користувачів довідково-інформаційної системи ресторану було вибрано мову програмування PHP. Так як PHP є вбудованою мовою то вона відрізняється винятковою гнучкістю по відношенню до потреб розробника.

Мова сценаріїв PHP Технологія Personal Home Page (PHP) має дуже широке поширення адже вона безкоштовна і підтримує найпопулярніші платформ. Вона базується на принципі побудови сторінок використовуючи шаблони, який вперше з'явився в Active Server Pages, але розвиває та доповнює його. Мова PHP має наступні характеристики: традиційність; простота, ефективність, безпека, гнучкість та багатофункціональність.

Велика кількість конструкцій мови запозичені з C і Perl. Це досить сильно знижує початкові зусилля у вивченні PHP.

Простота – сценарій PHP може включати в себе 10 000 рядків або складатися з одного рядка – все залежить від специфіки завдання. При цьому не потрібно довантажувати бібліотеки, вказувати їм спеціальні параметри компіляції. Механізм PHP починає опрацьовувати код після першої екрануючої послідовності та продовжує виконання до тих пір, поки він не знайде парну екранну послідовність. Якщо код має правильний синтаксис то він виконується так, як написав програміст.

Ефективність є досить важливим фактором при програмуванні для багатокористувацьких середовищ, куди входить і WWW. Створено механізм

									Арк.
									34
Змн.	Арк.	№ докум.	Підпис	Дата					



незалежною переносною мовою та вона може існувати на таких платформах, як UNIX, Solaris, FreeBSD і Windows [11].

PHP являється універсальною мовою для розробки сценаріїв, яку широко використовують. Спочатку вона була створена для розробки Web-вузлів. Розроблена Расмусом Лердофом (Rasmus Lerdof), PHP фактично використовувалася як засіб для створення користувачами своїх Web-сторінок (Personal Home Page). Але PHP виявився таким корисним і популярним, що досить швидко став повноцінною мовою програмування.

#### Переваги використання PHP:

- швидкодія Web-вузлів – оскільки код PHP вбудовується у HTML,
- час, який потрібний для використання, обробки і завантаження веб-сторінок суттєво зменшується;
- відкритість – PHP є доказом того, що безкоштовні версії існують;
- синтаксис – PHP достатньо простий для розуміння та забезпечує доступність використання. При цьому PHP розроблений так, щоб мати змогу бути легко вбудованим в HTML-сторінки;
- PHP може використовуватися під управлінням різних операційних систем, які забезпечують його універсальність, включаючи Windows, Linux, Mac OS та більшість систем сімейства Unix;
- всебічна технічна підтримка. Є офіційний Web-вузол PHP, де надається великий список тем для обговорення, який покриває різні питання;
- безпека – у випадку, якщо сценарій PHP розроблений правильно, то це означає, що його програмний код користувачі побачити не зможуть;
- зручні налаштування - відкритість PHP надає можливість модифікувати програмне забезпечення, додавати або змінювати його функціональність, необхідну для вирішення конкретних завдань.

PHP забезпечує достатній контроль над оточенням, дозволяючи зменшити ймовірність помилок. PHP – є універсальним засобом розробки сценаріїв загального призначення. Сценарії на мові PHP являють собою комп'ютерні

									ДППЗ.170107.01.07.ПЗ	Арк.
										36
Змн.	Арк.	№ докум.	Підпис	Дата						

файли, які містять створені на мові PHP інструкції, що виконують відповідні дії. Велика кількість сценаріїв включають послідовності інструкцій, які дозволяють виконувати завдання, від розробки веб-сторінок до навігації по файловій системі. Оскільки PHP створювалася для Web, вона включає багато можливостей, які призначені для використання в сценарії створення динамічних веб-сторінок. Для реалізації функціональних можливостей довідково-інформаційної системи «Ресторан» обрана мова PHP.

Також при розробці системи була використана мова розмітки гіпертексту Hyper Text Markup Language (HTML) яка призначена для написання гіпертекстових документів, що публікуються в World Wide Web. Гіпертекстовий документ – це текстовий файл, який має спеціальні мітки – теги, що надалі впізнаються браузером і використовуються ним для відображення вмісту файлу на екрані комп'ютера. За основу моделі розмітки документів в HTML використовується тегова модель. Тегова модель подає документ як сукупність контейнерів, де кожен з них починається і закінчується тегами. Тобто документ HTML є не чим іншим, як звичайним ASCII-файлом з доданими до нього керуючими HTML кодами (тегами). Теги HTML-документів як правило є простими і зрозумілими у використанні, оскільки вони створені за допомогою загальноживаних слів англійської мови, зрозумілих скорочень і позначень. Web-сторінки можуть знаходитись в будь-якому форматі, але як правило прийнято Hyper Text Markup Language – мова розмітки гіпертекстів, яка призначена для створення форматowanego тексту, який насичений зображеннями, звуком, анімацією, відео та гіпертекстовими посиланнями. HTML був ратифікований World Wide Web Consortium. Ця мова підтримується всіма браузерами. Так як HTML-документи записуються в ASCII I-форматі, то для його створення може бути використаний будь-який інший текстовий редактор. Основною перевагою гіпертексту є можливість додавати до вмісту документу гіперпосилань – спеціальні конструкції мови HTML, що дозволяють перейти до перегляду іншого документу. Графічна та звукова інформація, яка

										ДППЗ.170107.01.07.ПЗ	Арк.
											37
Змн.	Арк.	№ докум.	Підпис	Дата							

включається в HTML-документ, зберігається тут в окремих файлах. Програми перегляду HTML-документів (браузери) перетворюють теги розмітки і розташовують текст та графіку на екрані відповідним чином. HTML – це не мова програмування, вона служить тільки для розмітки сторінок, надання відповідного виду тому чи іншому елементу чи об'єкту. Відбувається це шляхом присвоєння кожному елементу своїх параметрів, що розпізнає браузер. Параметри такі можуть бути задані як для одного, так і для групи елементів. Мова гіпертекстової розмітки HTML – основний будівельний блок веб-сторінок, який використовується для створення та візуального представлення даних веб-сторінок. Вона визначає тільки зміст сторінки, але не її саму функціональність. Кожен може вибирати свій інструмент для створення Web-сторінок. Це, як правило, може бути MS FrontPage або Macromedia DreamWeaver, Allaire HomeSite або 1st Page. або Блокнотом (Notepad) [16].

Текстові редактори можна використовувати тільки для створення невеликих сторінок, тому що у них є багато мінусів: не підтримуються виконувани проекти, відсутнє "підсвічування" тексту, працювати вкрай незручно

Основним мінусом MS FrontPage є те, що він створює дуже великий HTML-код.

При здійсненні програмної реалізації інтерфейсів додатків користувачів довідково-інформаційної системи «Ресторан» використана мова розмітки сторінок HTML.

Також при розробці довідково-інформаційної системи була використана формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки – CSS – каскадні таблиці. CSS використовується для визначення кольорів, шрифтів та розташування відділених блоків, інших аспектів представлення зовнішнього вигляду даних Web-сторінок. Основна мета розробки CSS полягає в розділенні опису логічної структури від опису зовнішнього вигляду Web-сторінки. Такий поділ дозволяє збільшити доступність документа та надати велику гнучкість і можливість управління його

										ДППЗ.170107.01.07.ПЗ	Арк.
											38
Змн.	Арк.	№ докум.	Підпис	Дата							

поданням, а також зменшити складність та повторюваність в даному структурному вмісті. Роль CSS полягає в тому, що веб-дизайнери мали змогу створити єдину таблицю стилів для обслуговування всього веб-ресурсу. Всі сторінки HTML повинні виконувати єдиний набір даних інструкцій CSS, а код для HTML при цьому не буде перевантажений даними по форматуванню. Для того щоб таблиця стилів мала вплив на форматування документа, потрібно вказати браузеру на те, що вона існує, і де її розташовано. CSS може бути збережена у вигляді окремого документу і прив'язана потрібним чином до документу HTML, або повинна бути вбудована в HTML-документ. Можливе одномоментне використання декількох способів включення таблиць стилів в HTML документ.

CSS верстка сайтів користується попитом у сучасних веб-розробників. CSS – дозволяють веб-майстру достатньо гнучко задавати стилі відображення елементів на сторінках сайту та допомагають заощаджувати час і зусилля для верстки сайтів. Основні плюси CSS верстки у порівнянні з HTML версткою визначаються у більш зручному керуванні зовнішнім виглядом сторінок сайту та більш точному відображенні елементів незалежно від самого браузера, у більш зручному написанні коду та завдяки відділення вмісту сайту від його оформлення. Крім того, CSS верстка досить істотно розширює стандартні можливості HTML та дозволяє їй представляти один і той же документ в різних стилях або методах його виведення, таких як екранне уявлення, друковане уявлення та читання голосом.

#### Основні переваги CSS:

- більш чистий код;
- такий код легше підтримувати;
- він швидше завантажується;
- краще оптимізований для використання в пошукових системах;
- модульний код;

									ДППЗ.170107.01.07.ПЗ	Арк.
										39
Змн.	Арк.	№ докум.	Підпис	Дата						

- правила стилю також можуть застосовуватися і до безлічі сторінок;
- однаковий дизайн.

CSS верстка сайтів досить часто може застосовуватися для самих різних за тематикою і структурою сайтів, вона сумісна з більшістю сучасних браузерів. CSS верстка сайтів дозволяє досить гнучко управляти зміною властивостей цих елементів при взаємодії з користувачами сайтів.

Для реалізації довідково-інформаційної системи «Ресторан» та зберігання всієї потрібної інформації була обрана система управління базою даних MySQL. Використання цієї СУБД дозволяє створити бази даних обраної предметної області, згідно з нашими вимогами. СУБД MySQL – це невеликий та компактний багатопотоковий сервер баз даних, що характеризується великою швидкістю, стійкістю та легкістю у використанні. СУБД MySQL розроблена компанією ТсХ для своїх внутрішніх потреб, що полягали у швидкій обробці досить великих баз даних. MySQL є гарним вибором для малих і середніх додатків. Вихідні тексти для сервера компілюються на безлічі платформ. Найбільш повно такі можливості сервера проявляються на Unix-серверах, там де є підтримка багатоканальності яка дає значний приріст продуктивності.

Основні переваги у застосування СУБД MySQL:

- підтримується необмежена кількість користувачів, які одночасно працюють з базою даних;
- кількість рядків в таблицях може бути 50 мільонів;
- виконується швидке опрацювання команд;
- проста та ефективна система безпеки.

MySQL – це швидкий сервер, але для того щоб досягти цього розроблювачам довелося пожертвувати деякими вимогами до реляційних СУБД. Тому в MySQL немає: підтримки вкладених запитів; підтримки зовнішніх ключів; підтримки тригерів та збережених процедур; підтримки уявлень view. Саме така відсутність підтримки транзакцій та зовнішніх ключів, тригерів та збережених процедур дали можливість досягти досить високої швидкодії. Ці

						ДППЗ.170107.01.07.ПЗ	Арк.
							40
Змн.	Арк.	№ докум.	Підпис	Дата			

можливості не є такими критичними при створенні Web-додатків, що разом з високою швидкістю і малою ціною дозволило одразу стати популярним. Крім того, СУБД MySQL має досить хорошу підтримку.

MySQL забезпечена розширеним довідковим керівництвом, можна укласти контракти для технічної підтримки з самими розробниками MySQL. Існує такий список розсилки, від грамотних користувачів, в тому числі включаючи і самих розробників MySQL.

MySQL – це вільно поширювана СУБД, яка має клієнт-серверну архітектуру. До сервера MySQL також можуть звертатися різні клієнтські програми, в тому числі і з віддалених комп'ютерів. MySQL – це СУБД з відкритим кодом. Будь-який бажаючий тут може безкоштовно завантажити програмні коди на сайті розробника та при необхідності доопрацювати її. Існує багато додатків MySQL, які створені і вільно розповсюджені сторонніми розробниками. Але для застосування MySQL в платному додатку необхідно придбати комерційну ліцензовану версію цієї програми у компанії MySQL AB. MySQL – кросплатформова система. Її можна використовувати безперечно у всіх сучасних операційних системах, в тому числі і у Windows, Linux, Mac OS, Solaris. Так само вона має безліч програмних інтерфейсів, завдяки яким до бази даних MySQL можуть завжди підключатися додатки, створені за допомогою C / C ++ , Java, Perl, PHP, Python, Tcl, ODBC, NET і Visual Studio. MySQL має досить гарні технічні характеристики: багатоканальність, розраховану на доступ багатьох користувачів, швидкість, масштабованість. MySQL включає в себе розвинену систему забезпечення безпеки та розмежування доступу на основі системи механізму привілеїв. MySQL є реляційною СУБД, це означає, що дані в її базах зберігаються у вигляді логічно пов'язаних між собою таблиць, для доступу до яких потрібно здійснювати запитів за допомогою SQL.

Великим плюсом MySQL є можливість роботи з інтерфейсом програмного додатка API (Application Program Interface). API має можливість забезпечити простий доступ від програми користувача до СУБД. Якщо навіть ці програми

									Арк.
									41
Змн.	Арк.	№ докум.	Підпис	Дата					

ДППЗ.170107.01.07.ПЗ

будуть написані не на Perl чи C.. Найпопулярнішою тандемом для управління сайтами вважається MySQL з мовою PHP. Багато CMS створені на PHP у взаємодії з базою даних MySQL. Одним із вдалих прикладів цього «союзу» може бути движок для 30 сайтів і блогів WordPress, що завоював широку популярність у світі. Взаємодія з MySQL в такому випадку ведеться за допомогою сукупності функцій. Прикладом такої функції може бути `mysql_connect`, яка з'єднується з сервером баз даних і повертає дескриптор з'єднання з нею. Існує безліч СУБД, що підтримують мову запитів SQL: MySQL, MS SQL, PostgreSQL, MSSQL.. Кожна з них має кращі сторони в певній сфері. І все ж саме MySQL завоювала широке визнання та популярність в Інтернеті завдяки своїй багатогранності та універсальності.

У цьому розділі було проведено проектування довідково-інформаційної системи, а саме архітектури. Було розроблено структуру довідково-інформаційної системи та спроектовано базу даних та інтерфейс користувача. Був наданий аналіз та проведено вибір технологій та методів реалізації програмного забезпечення.

					ДППЗ.170107.01.07.ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Структура і функціональне призначення модулів системи, їх взаємозв'язок.

#### Наповнення сайту

Всі данні веб-сайту зберігаються в таблицях бази даних. Розглянемо деякі з них. Спочатку наведемо опис самих таблиць даної бази даних.

Таблиця 3.1. Опис таблиць бази даних

Ім'я таблиці	Опис
posts	Головна таблиця бази в де зберігаються всі статі і необхідна службова інформація
postmeta	Додаткові данні до публікацій, що відсутні в стандартних полях у таблиці posts.
users	Відомості користувачів, що є авторами статей
usermeta	Додаткові поля необхідні для опису користувачів та доповнення до таблиці users.
comments	Зберігаються коментарі для створених статей.
terms	Перелік рубрик, до яких відноситься стаття..
term_taxonomy	Таблиця призначена для систематизації рубрик
term_relationships	Вказується відповідність статті до рубрики
links	Зберігається інформація про посилання
options	Всі налаштування движка.

Проаналізуємо зміст таблиці «Збереження записів, чернеток, бібліотеки медіа файлів».

Таблиця 3.2. Зміст таблиці «Збереження записів, чернеток, медіа файлів»

					ДППЗ.170107.01.07.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43





Кінець таблиці 3.4

1	2
user_pass	Пароль користувача в закодованому виді.
user_nicename	Ім'я для звернення до користувача. Воно збігається з ім'ям входу.
user_email	Електронна пошта для зв'язку .
user_url	Адреса сайту користувача.
user_registered	Часреєстрації користувача.
user_activation_key	Ключ для активації.
display_name	Ім'я для відображення.

Ігноруючи існування таблиці users, всі пов'язані з користувачем дані знаходяться в таблиці usermeta. Її зміст описаний в наступній таблиці

Таблиця 3.5. Вміст таблиці usermeta

Поле	Опис
umeta_id	Ідентифікатор метазапису, первинний ключ.
user_id	Ідентифікатор користувача, що відноситься до запису.
meta_key	Запасний параметр
meta_value	Значення запасного параметру.

Переглянемо вміст наступної таблиці comments

Таблиця 3.6. Вміст таблиці comments

Поле	Опис
1	2
comment_ID	Ідентифікатор коментарів та первинний ключ.
comment_post_ID	Ідентифікатор коментарів



дозволяє людям, що не особливо добре розуміються в html створювати легенькі сторінки, а іншим – збільшити їх функціональність та можливості при редагуванні коду вручну. Можна закодувати материнську сторінку, в яку будуть вкладені підсторінки, або скористатися функцією "Меню" і вставити на головній сторінці каталоги, де будуть розміщені всі інші. Використовуючи функцію "Меню" і публікації сторінок, в довідково-інформаційній системі «Ресторан» сторінки були заповнені усілякою інформацією та файлами.

### 3.2 Інструкція користувача

Розглянемо покроково як користуватись довідково-інформаційною системою «Ресторан» та на які блоки вона розбита. Перш за все після того як ми заходимо в нашу довідково-інформаційну систему «Ресторан» ми бачимо її головну сторінку.

На ній розташована назва, страви дня, кнопка для можливості зайти і переглянути , що можна замовити.



Рисунок 3.1 – Головне меню довідково-інформаційної системи «Ресторан»

Також на основній сторінці є можливість пройти реєстрацію та авторизацію.

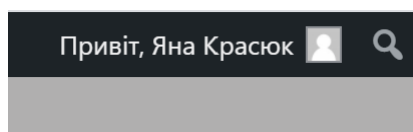


Рисунок 3.2 – Реєстрація

Якщо скористатись кнопкою прокрутки то далі на основній сторінці ми побачимо інформацію про сам ресторан. Тут буде перелік статей про позитивні сторони ресторану, його концепцію та місце на ринку серед інших закладів харчування. Можна побачити фото ресторану. Головна особливість даної частини сторінки в тому що розміщення інформації наче «накладається» на різні гарні фото з ресторану, що сприяє позитивному сприйняттю інформації про заклад, а це в свою чергу збільшує відвідини даної довідково-інформаційної системи відповідно сприяє більш активному не тільки перегляду але і замовленню страв.

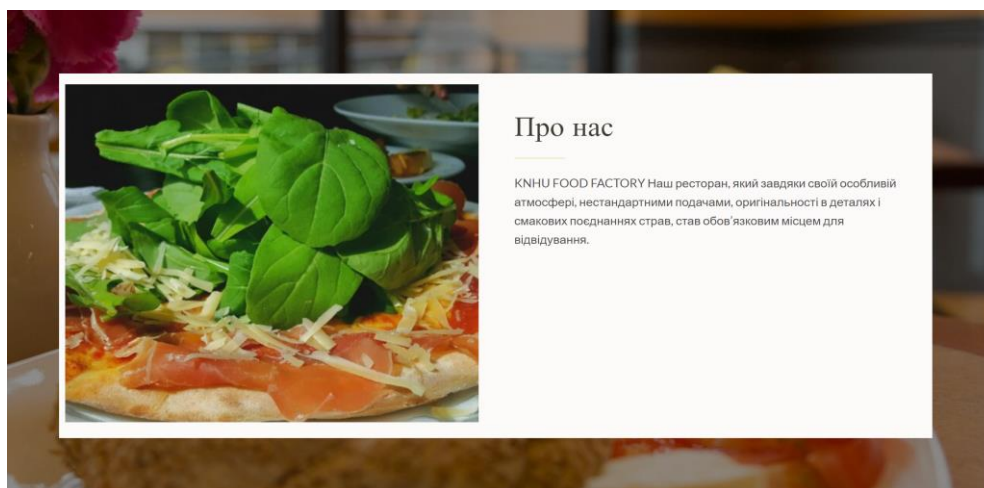


Рисунок 3.3 – Фрагмент основної сторінки

А далі після прокрутки буде розміщена інформація про місце розташування самого закладу та його додаткові позитивні сторони.

					ДППЗ.170107.01.07.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

## Ми пропонуємо

Друзі! Навіть карантин в не завадить вам насолодитися улюбленими стравами! Відтепер наш заклад працюватиме з 10 по 22 на виніс за спеціальним меню!

### 🕒 Відкриті

Замовити їжу через вікно торгівлі або зайти до закладу та замовити страви на виніс..

### 🚗 Вільний паркінг

Ресторан у центрі міста з вільним паркінгом.

### 📍 Розташування в центрі міста

Ми розташовані в історичній частині міста, за 5 хвилин ходьби від площі незалежності



Рисунок 3.4 – Фрагмент головної сторінки

## Наше меню

м'ясне, рибне, вегетаріанське, карта вин, карта коктейлів, десерти.

Салати    Основні страви    Десерти    Перші страви



### Сирний крем суп

€45.00

Ніжна текстура сирного крем-супу захопає вас з першої ложки.

[Додати у кошик](#)



### Салат Рікота

€140.00

Свіжі овочі: морква, огірок, селера, солодкий перець з сиром фета у ніжному соусі Рікота

[Додати у кошик](#)



### Салат від шеф-кухаря

€270.00

Ніжна, із злегка терпким горіховим смаком, приємно-зелений м'якуш авокадо надасть смаку неповторність, а м'ясо тунця окрім вишуканого ніжного смаку не лише є легким поживним продуктом, але і цінується за насиченість білком і гемоглобіном.

[Додати у кошик](#)



### Класичний салат Кобб

€250.00

салат Кобб готується з обсмаженого бекону і курячого філе, зварених круто яєць – це та частина інгредієнтів, яка вимагає термообробки. Перепелині яйця можна замінити на пару курячих. овочева частина – не викличе складнощів.

[Додати у кошик](#)





Рисунок 3.5 – Фрагмент головного меню

Далі рухаючись по «Наше меню» можна вибрати для замовлення М'ясне, рибне, вегетаріанське меню, переглянути карту вин, коктейлів та вибрати десерт. Після того як все це ми додамо у кошик перейдемо туди.

										Арк.
										50
Змн.	Арк.	№ докум.	Підпис	Дата						

ДППЗ.170107.01.07.ПЗ

### Кошик

	Товар	Ціна	Кількість	Проміжний підсумок
×	 Салат Рікота	€140.00	<input type="text" value="2"/>	€280.00
×	 Класичний салат Кобб	€250.00	<input type="text" value="2"/>	€500.00
×	 Зелений салат	€80.00	<input type="text" value="1"/>	€80.00
×	 Сирний крем суп	€45.00	<input type="text" value="1"/>	€45.00

Код купону

#### Підсумки кошика

Проміжний підсумок	€905.00
Загалом	€905.00

Рисунок 3.6 – Фрагмент сторінки «Кошик»

На сторінці «Кошик» можна переглянути вибрані страви, їх кількість та загальну суму замовлення та застосувати купон на знижку при його наявності.

Потім при натисненні кнопки «Перейти до оформлення» переходимо на сторінку «Оформлення».

Додатково кожне блюдо описується окремо і додається відгук усіх зареєстрованих користувачів.

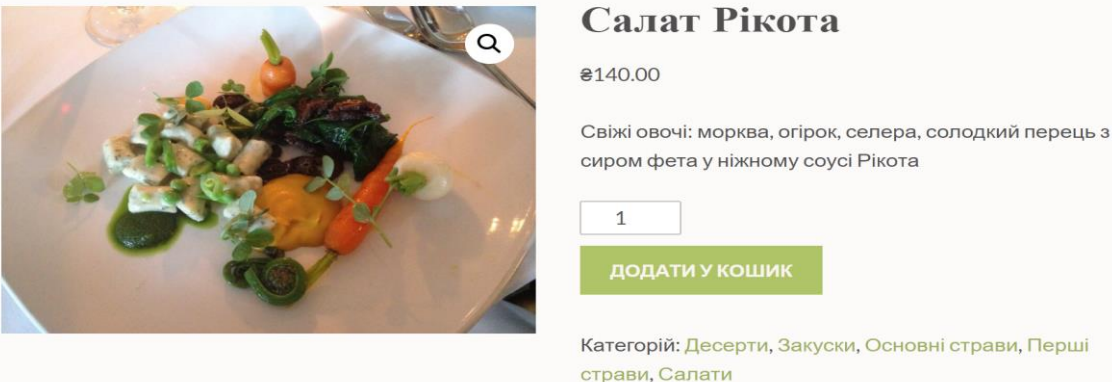
### Оформлення

Маєте купон знижки? Натисніть тут, щоб ввести код купону знижки

Платіжні дані	Додаткова інформація
<p>Ім'я *</p> <input type="text" value="Яна"/>	<p>Примітки до замовлень (необов'язково)</p> <p>Нотатки до вашого замовлення, наприклад спеціальні примітки для доставки.</p>
<p>Прізвище *</p> <input type="text" value="Красюк"/>	
<p>Назва компанії (необов'язково)</p> <input type="text"/>	
<p>Країна/Регіон *</p> <input type="text" value="Україна"/>	

Рисунок 3.7 – Фрагмент сторінки «Оформлення»

На цій сторінці ми заповнюємо всі свої контактні дані, якщо користувач не зареєстрований і проводимо оплату. Користувачу потрібно надати, якщо є купон на знижку та додати коментар по своєму замовленню, якщо є якісь зауваження щодо наповнення блюд чи побажань клієнта. В тих даних які потрібно заповнити є пункт «Назва компанії» це невеличке нововведення для того щоб корпоративні клієнти або користувачі з одного закладу могли зробити групове замовлення, що досить цікаво і для людей і для закладу. Заклад – зможе здешевити доставку і тим самим збільшити кількість клієнтів, а замовникам зробити групове замовлення і тим самим отримати знижку.



**Салат Рікота**  
 €140.00  
 Свіжі овочі: морква, огірок, селера, солодкий перець з сиром фета у ніжному соусі Рікота  
 1  
 ДОДАТИ У КОШИК  
 Категорій: Десерти, Закуси, Основні страви, Перші страви, Салати

ОПИС    ВІДГУКИ (0)

**Опис**  
 Свіжі овочі: морква, огірок, селера, солодкий перець з сиром фета у ніжному соусі Рікота

Рисунок 3.8 – Фрагмент вікна «Салат Рікота»

Одночасно можна з цієї сторінки зайти в інші категорії страв, такі як Десерти, Закуси, основні страви, Перші страви, Салати, що дуже зручно. Не потрібно кожного разу виходити на головну сторінку і там вже вибирати потрібну категорію.

						ДППЗ.170107.01.07.ПЗ	Арк.
							52
Змн.	Арк.	№ докум.	Підпис	Дата			

### 3.4 Технічні характеристики програмного забезпечення

Для того, щоб система правильно та якісно працювала потрібно, щоб комп'ютер відповідав необхідним характеристикам:

- тактова частота процесора – не нижче 1 ГГц ;
- об'єм ОП – не менше 1024 Мбайт;
- об'єм вільного місця на жорсткому диску - не менше 400 Мбайт;
- мінімальний об'єм відео пам'яті – 512 Мбайт;
- операційна система – Windows (Windows 7, або пізніша версія до Windows 10);
- розширення екрану 1024 x 768
- наявність CD/DVD – ROM (для встановлення);
- периферія (клавіатура, миша).

Також для запуску на ПК потрібно Framework 4.5 або вище, для забезпечення функціонування Windows додатків, створених на Microsoft Visual Studio. Даний пакет постачається із більшістю сучасних версій Windows, та є стандартним, але в окремих випадках потрібно встановлювати цю версію.

У даному розділі була проведена реалізація бази даних та модулів системи. Розроблена інструкція користувача. Детальний код програми описаний у додатку Б.

									ДППЗ.170107.01.07.ПЗ	Арк.
										53
Змн.	Арк.	№ докум.	Підпис	Дата						

## 4 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Вибір та обґрунтування методів тестування системи

Сучасний світ не можливо уявити без мережі Інтернет та величезної кількості web-сайтів та web-сторінок. Кількість розробок, які доступні через браузер, збільшується досить швидко, тому з метою економії часу та людських ресурсів, може виникнути необхідність тестувати такі проекти автоматично. Такий засіб тестування є більш точним, тому що виключає людський фактор, є значно швидшим, та має ширші можливості аніж ручне тестування. Основне мета автоматизованого web-тестування – емуляція дій користувача, пошук помилок, використання різних вхідних даних, спостереження за поведінкою системи коли збільшується навантаження на неї.

Тестування програмного забезпечення – це процес, в якому проводиться перевірка якості програмного забезпечення на відповідність до заявлених вимог. Як правило, поняття якості обмежено такими термінами, як безпечність, коректність, повнота, але може включати більш технічні вимоги, які приведені в стандарті ISO 9126 [26].

Тест – це спеціальна, штучно створена ситуація, яка обрана певним чином, та подальший опис того, які спостереження за роботою програми потрібно зробити для перевірки її відповідності деяким вимогам.

Тест-кейс є найменшим компонентом тест-сценарію, але при цьому він також часто включає в себе декілька кроків. Наприклад, при тестуванні продуктивності штучно створена подія може проходити протягом тривалого часу навантаження на систему. Спостереження, які при цьому роблять – це набір різних графіків або метрик, за допомогою яких тестувальник робить певні висновки. Таким чином, можна зрозуміти, що тестувальник робить в процесі тестування дві дії:

– по-перше, керує виконанням програми та створює штучні тестові ситуації, за допомогою яких перевіряється поведінка програми;

									ДППЗ.170107.01.07.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата						54

– по-друге, спостерігає за поведінкою програми і проводить порівняння дійсного результату з очікуваним.

Під час створення автоматизації тестування тестувальник не спостерігає сам за поведінкою програми, а делегує це завдання спеціальному інструменту або спеціальній, до цього написаній програмі. В такому випадку програма проводить спостереження, як відбувається порівняння поведінки з очікуваним, а тестувальник проводить аналіз лише деякого кінцевого результату – чи збігається поведінка, що спостерігається з очікуванням, або ні.

Виділяють такі основні методи тестування: модульне тестування, інтеграційне тестування, прийомне тестування та тестування системи вцілому.

Програмне забезпечення піддається випробуванню в такому порядку: модульне тестування; інтеграційне тестування; системне тестування; прийомні випробування. В першу чергу проводиться модульний тест. Це метод випробування на об'єктному рівні. Відомі програмні компоненти тестуються на наявність помилок. Для цього тесту необхідно знання програми та кожного встановленого модуля. Таким чином, ця перевірка здійснюється програмістами, а не тестувальниками. Для цього створюються тест-коди, які перевіряють поведінку програмного забезпечення так, як очікувалось. Під час інтеграційного тестування відомі модулі, які вже були протестовані модульно, інтегруються один з одним, перевіряються на наявність несправностей. Таке тестування в першу чергу виявляє помилки інтерфейсу. В системному тестуванні вся система перевіряється на наявність помилок і багів. Цей тест здійснюється шляхом сполучення апаратних та програмних компонентів всієї системи, а потім виконується її перевірка. Це тестування відоме також під методом тестування "чорної скриньки", де перевіряються очікувані для користувача умови роботи програмного забезпечення. Прийомні випробування – це останній тест, який проводиться перед передачею програмного забезпечення клієнту. Він проводиться, щоб гарантувати, що програмне забезпечення, яке було розроблено, відповідає всім вимогам замовника.

										Арк.
										55
Змн.	Арк.	№ докум.	Підпис	Дата						

Розглянемо основні переваги автоматизованого тестування. Автоматизація тестування відкриває багато можливостей, і, зокрема, дозволяє створювати якісні програми з меншими зусиллями. Але багато компаній все ще віддають перевагу ручному тестуванню, через брак знань та спеціалістів, які могли б краще інтегрувати автоматизацію тестування в процес розробки.

Автоматизація тестування передбачає, що машина буде виконувати те, що робила людина, при цьому набагато швидше, з меншою ймовірністю помилки і безперервно. Люди схильні швидко втрачати концентрацію, роблячи таку роботу, і упускати наявні помилки і недоліки. В цьому відношенні машини набагато швидші і надійніші.

Звичайно, у автоматизації є і недоліки. Цей процес не усуне специфічні проблеми, з якими можуть зіткнутися користувачі. Ще один недолік – обмежене число тестових сценаріїв. Головне завдання автоматизації – знайти баги в простих операціях, наприклад, вхід в додаток, створення аккаунта або відправка електронного листа, коли користувач забуває пароль. Недоліки додатку, що виявляються при специфічних обставинах, все ще повинні тестуватися вручну. Як відомо, машини мають більші можливості, але, на жаль, вони не можуть думати самостійно. Крім того, автоматизація не дозволяє провести ефективне тестування дизайну, наприклад, положення кнопки, і, в цілому, наскільки зручний додаток у використанні. Для з'ясування всього цього все ще застосовується ручне тестування.

Сьогодні тестування може бути автоматизоване практично для всіх видів програмного забезпечення, яке розроблюється. Тестування web-додатків також стало важливою і великою частиною процесу автоматизації тестування. Інтернет увійшов у життя людей давно і став його частиною. Різні веб-додатки потрібно тестувати, тому що погано написаним і не протестованим додатком користуватися ніхто не буде. Більш того помилки в таких додатках можуть призвести до незручностей для користувачів, інколи до втрати грошей чи важливої інформації. Для забезпечення якості створених web-додатків,

										ДППЗ.170107.01.07.ПЗ	Арк.
											56
Змн.	Арк.	№ докум.	Підпис	Дата							

тестувальниками розробляється цілий набір різноманітних тестів, включаючи функціональне тестування, юзабіліті-тестування, кросбраузерне тестування а також навантажувальне та стрес-тестування [29].

#### 4.2 Розробка тестових наборів даних

До актуальних та нетривіальних задач тестування слід віднести складання тестових сценаріїв, які брали б до уваги специфічні вимоги до Web-додатків та дозволяли б ефективно виявляти помилки в розроблених кодах. Одним з різновидів автоматичного тестування є тестування на основі моделей, при якому варіанти тестування будуються із деякої моделі web -додатку, що тестується. Рисунок 4.1 відображає загальне представлення процесу тестування на основі моделей. Згідно з вимогами до web-додатку, здійснюється розроблення моделі, яка відображає ті особливості системи, що підлягають тестуванню. Ця модель дозволяє генерувати набори варіантів сценаріїв тестування, які визначають умови, вхідні дані, очікувані результати роботи системи та оцінювання результатів тестування з використанням компаратора.

Порівняння в компараторі очікуваних та реальних результатів дає можливість зробити висновок про наявність помилок в системі, надійність та відповідність системи моделі згідно з вимогами до неї. Крім того, може бути зроблено висновок про необхідність модифікації існуючої моделі чи додаткової генерації варіантів тестування.

Ще однією особливістю тестування web-додатків є необхідність зберігати інформацію про стан сесії користувача.

					ДППЗ.170107.01.07.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

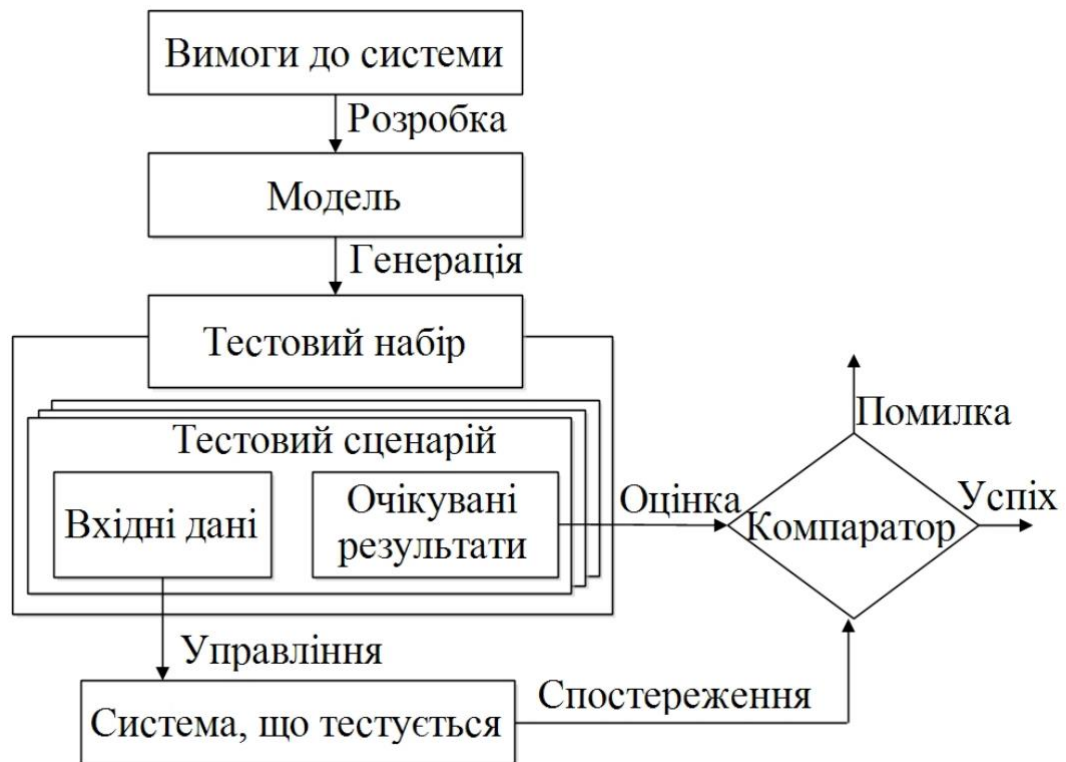


Рисунок 4.1 - Тестування на основі моделі web-додатку

В результаті цього тестовий сценарій має враховувати послідовність тестових випадків, тобто декількох дій в системі.

Таблиця 4.1 – Сценарій для тестування інтерфейсу користувача

Дія	Реакція програми	Результат
1	2	3
1. Натиснути на кнопку «Салати»	З'являється опис інгредієнтів салату та їх фото	Вірно

Кінець таблиці 4.1

1	2	3
2. У діалоговому вікні ввести ім'я користувача Яна Красюк та пароль 11111. Далі натиснути кнопку «Вхід»	Вікно-«Вхід» повинно закритися і з'явиться «Привіт, Яна Красюк»	Вірно
3. Зайти а головному меню в меню «Супи»	Повинна відкритись сторінка з переліком супів	Не вірно. Відкрилась сторінка «Салати»

Після цього було проведено навантажувальне тестування, щоб виявити як буде себе вести дана довідково-інформаційна система під час великого напливу людей для замовлень та перегляду.

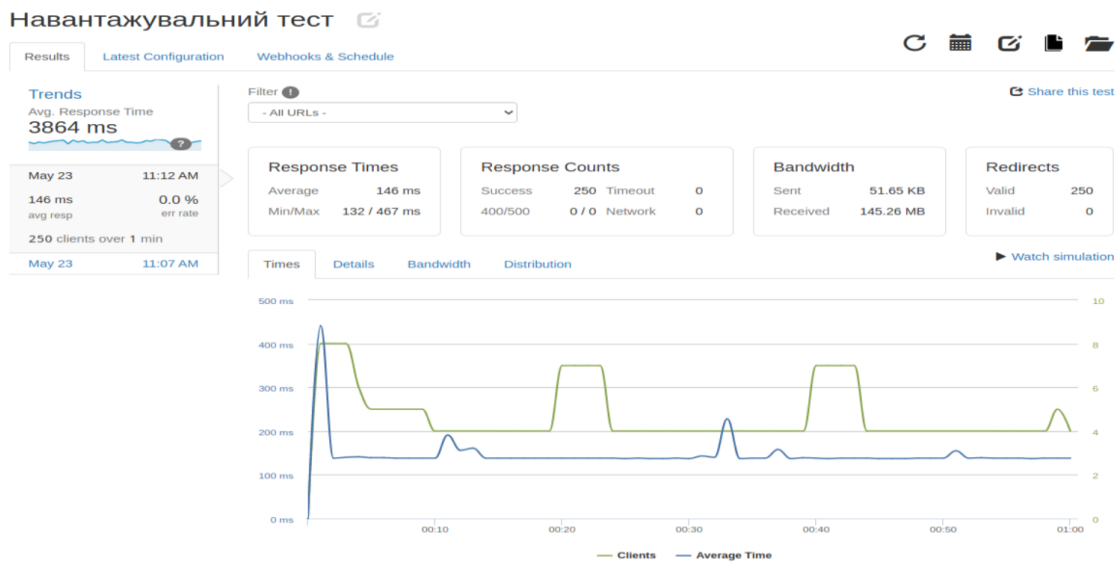


Рисунок 4.2 – Результати навантажувального-тесту

Наша довідково-інформаційна система була протестована за допомогою спеціальної програми, яка виявила, що навантажувальне тестування програма успішно пройшла. Про що свідчить графік, який зображає весь процес.

В цьому розділі проводиться налагодження та тестування системи з вибором та обґрунтуванням методів тестування, аналіз результатів тестування.

					ДППЗ.170107.01.07.ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

Отже, вході виконання курсового проекту було виконано заплановане завдання, а саме розроблено довідково-інформаційну систему «Ресторан».

Було проведено дослідження предметної області та виділення функціональних та нефункціональних вимог до розроблюваної довідково-інформаційної системи та здійснено постановку задачі. Було проведено аналіз існуючих довідково-інформаційних систем та дана їх характеристика.

Проведено проектування довідково-інформаційної системи, а саме архітектури. Було розроблено структуру довідково-інформаційної системи та спроектовано базу даних та інтерфейс користувача. Був наданий аналіз та проведено вибір технологій та методів реалізації програмного забезпечення.

Була розроблена реалізація бази даних та модулів системи. Розроблена інструкція користувача, для зручного використання сайту. Також проводилося налагодження та тестування системи з вибором та обґрунтуванням методів тестування, аналіз результатів тестування.

					ДППЗ.170107.01.07.ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Задачи автоматизации ресторана, кафе, бара [Электронный ресурс]: CNews Клуб // <http://club.cnews.ru> URL: [http://club.cnews.ru/blogs/entry/zadachi\\_avtomatizatsii\\_restorana\\_kafe\\_bara](http://club.cnews.ru/blogs/entry/zadachi_avtomatizatsii_restorana_kafe_bara) (дата обращения: 16.04.2019).

2. 1С:Предприятие 8. Ресторан [Электронный ресурс] : 1С URL: <https://solutions.1c.ru/catalog/restaurant/features> (дата обращения: 16.04.2019).

3. Система управления торговым предприятием «S-MARKET» [Электронный ресурс]: ДатаКрат URL: <https://www.datakrat.ru/solutions/17682.html> (дата обращения: 16.04.2019).

4. Dynamics AX — теперь Dynamics 365 for Finance & Operations | Microsoft [Электронный ресурс]: Microsoft URL: <https://dynamics.microsoft.com/ru-ru/ax-overview/> (дата обращения: 16.04.2019).

5. Restaurant Head Office Management Software [Электронный ресурс]: NFS URL: <https://www.nfs-hospitality.com/solutions/restaurant-epos/enterprise/> (дата обращения: 16.04.2019).

6. MarketMan Software [Электронный ресурс] : Software Advice URL: <https://www.softwareadvice.com/retail/marketman-profile/> (дата обращения: 16.04.2019).

7. Head Office Module [Электронный ресурс]: POSist URL: <https://www.posist.com/restaurant-enterprise-software/head-office-management/> (дата обращения: 16.04.2019).

8. iikoChain: автоматизация ресторанного бизнеса, сети [Электронный ресурс]: Рестоучет URL: <http://www.restouchet.ru/iiko/about-iikochain/> (дата обращения: 16.04.2019).

9. Что такое SPA или одностраничный портал [Электронный ресурс]: CodeNet URL: <http://www.codenet.ru/webmast/js/spa/> (дата обращения: 16.04.2019).

10. Top JavaScript Frontend Frameworks Comparison in 2018 [Электронный

										Арк.
										62
Змн.	Арк.	№ докум.	Підпис	Дата						

ресурс]: FusionCharts URL: 83 <https://www.fusioncharts.com/resources/developers/js-frontend-frameworkscomparison> (дата обращения: 16.04.2019).

11. Что такое Angular. Начало работы с фреймворком [Электронный ресурс]: Metanit URL: <https://metanit.com/web/angular2/1.1.php> (дата обращения: 16.04.2019).

12. UML — диаграмма вариантов использования (use case diagram) [Электронный ресурс]: Хабрахабр // <https://habrahabr.ru> URL: <https://habrahabr.ru/post/47940/> (дата обращения: 16.04.2019).

13. Какие бывают требования? [Электронный ресурс]: UML2.ru Сообщество аналитиков // <http://www.uml2.ru> URL: <http://www.uml2.ru/faq/faqrequirements/34/> (дата обращения: 16.04.2019).

14. Норман Д., Дизайн привычных вещей // Манн, Иванов и Фербер, 2018, 384 с.

15. Интерфейс — зло [Электронный ресурс]: Бюро Горбунова // <https://bureau.ru> URL: <https://bureau.ru/bb/ui/evil/> (дата обращения: 16.04.2019).

16. Exploring EcmaScript Decorators [Электронный ресурс]: Google Developers // <https://medium.com> URL: <https://medium.com/googledevelopers/exploring-es7-decorators-76ecb65fb841> (дата обращения: 16.04.2019).

17. A quick intro to Dependency Injection: what it is, and when to use it [Электронный ресурс]: freeCodeCamp // <https://medium.com> URL: <https://medium.freecodecamp.org/a-quick-intro-to-dependency-injection-what-it-is-and-when-to-use-it-7578c84fa88f> (дата обращения: 16.04.2019).

18. Angular – Architecture Overview [Электронный ресурс]: Angular Docs // <https://angular.io> URL: <https://angular.io/guide/architecture> (дата обращения: 16.04.2019).

19. How to define a highly scalable folder structure for your Angular project [Электронный ресурс]: ITNEXT // <https://medium.com> URL: <https://itnext.io/choosing-a-highly-scalable-folder-structure-in-angular-d987de65ec7> (дата обращения: 16.04.2019).

										ДППЗ.170107.01.07.ПЗ	Арк.
											63
Змн.	Арк.	№ докум.	Підпис	Дата							

20. An introduction to plotly.js — an open source graphing library [Электронный ресурс]: freeCodeCamp // <https://medium.com> URL: <https://medium.freecodecamp.org/an-introduction-to-plotly-js-an-open-sourcegraphing-library-c036a1876e2e> (дата обращения: 16.04.2019).

21. Рынок доставки еды в картинках и в цифрах [Электронный ресурс]: Городская еда // <http://gorodskayaeda.ru> URL: <https://gorodskayaeda.ru/news/25161-rynok-dostavki-edy-v-kartinkah-i-v-tsifrah/> (дата обращения: 16.04.2019).

22. The changing market for food delivery [Электронный ресурс]: McKinsey // <https://www.mckinsey.com> URL: <https://www.mckinsey.com/industries/high-tech/our-insights/the-changing-marketfor-food-delivery> (дата обращения: 16.04.2019).

23. Технологии еды. Как и почему изменится доставка из ресторанов [Электронный ресурс]: Forbes // <https://www.forbes.ru> URL: <https://www.forbes.ru/biznes/363613-tehnologii-edy-kak-i-pochemu-izmenitsyadostavka-iz-restoranov> (дата обращения: 16.04.2019).

24. r\_keeper v.7 [Электронный ресурс]: U\_C\_S // <https://www.ucs-spb.ru> URL: <https://www.ucs-spb.ru/programms/r-keeper-v-7/> (дата обращения: 16.04.2019).

25. Система автораздачи Drive Thru [Электронный ресурс]: Группа компаний ККС // <https://ccrs.ru/> URL: <https://ccrs.ru/solutions/avtorazdacha/> (дата обращения: 16.04.2019).

26. RK-Order [Электронный ресурс]: EFACTORY // <http://efactory.ru> URL: <http://efactory.ru/service-rkorder.php> (дата обращения: 16.04.2019).

27. Как заполнять бизнес-модель Остервальдера: потребительские сегменты [Электронный ресурс]: Нетология // <https://netology.ru/> URL: <https://netology.ru/blog/ostervalder-segmenty> (дата обращения: 16.04.2019).

28. Охрана труда [Электронный ресурс] / Безопасность жизнедеятельност. URL: <http://www.grandars.ru/shkola/bezopasnostzhiznedeyatelnosti/ohrana-truda.html>, свободный (дата обращения: 05.05.2018).

										ДППЗ.170107.01.07.ПЗ	Арк.
											64
Змн.	Арк.	№ докум.	Підпис	Дата							

29. СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.

					ДППЗ.170107.01.07.ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК А  
(обов'язковий)

**ТЕХНІЧНЕ ЗАВДАННЯ**

## **Введення**

Робота виконується відповідно до проекту довідково-інформаційної системи Ресторан.

### **1. Підстава для розробки**

Підставою для розробки є «Завдання на дипломний проект» затверджене завідувачем кафедри інженерії програмного забезпечення.

Найменування розробки: Довідково-інформаційна система Ресторан

### **2. Призначення розробки**

Довідково-інформаційна система Ресторан призначена для оптимізації роботи закладу громадського харчування, його популяризації в мережі Інтернет, та покращенню зв'язку між клієнтами та рестораном.

Користувачами програми є відвідувачі та працівники ресторану.

Основною метою даного дипломного проекту є створення довідково-інформаційної системи «Ресторан». Ця система повинна бути забезпечена зручним та зрозумілим інтерфейсом, в якому буде розглянуто такі розділи як:

- меню
- контактна інформація
- відгуки
- страва дня
- опис блюда з фото
- можливість замовити он-лайн з доставкою
- здійснення оплати платіжною картою
- зміна мови
- години роботи

Відвідувачі повинні мати змогу в любий час переглянути меню ресторану, склад блюд, відгуки, чи працює в даний час заклад. Якщо в даний час ресторан відкритий то мати можливість зробити замовлення, оплату платіжною картою.

У програмі необхідно передбачити: можливість коригування налаштувань системи; резервне збереження даних; можливість зміни пароля входу в систему;

наявність вбудованої довідкової системи; швидкий пошук необхідних документів та довідкової інформації.

### **3. Вимоги до надійності**

Розроблюване ПЗ повинно мати: – можливість самовідновлення після збоїв (відключення електроживлення, збої в операційній системі тощо); – парольний захист при запуску програми; обмеження несанкціонованого доступу до даних; можливість резервного копіювання інформаційної бази; розмежування прав користувача; виключення несанкціонованого тиражування програми; контроль інформації, що вводиться, та блокування некоректних дій користувача при роботі з системою.

### **4. Вимоги до складу та параметрів технічних засобів**

Системні вимоги для роботи програмного забезпечення повинні бути наступними: тактова частота процесора – 1200 Гц; обсяг оперативної пам'яті – 64 Мб; обсяг вільного дискового простору – 50 Мб; роздільна здатність монітора 1024×768; наявність пристрою читання компакт-дисків. 3.4 Вимоги до інформаційної та програмної сумісності Програма повинна працювати в операційних системах Windows XP/7/10. Всі звіти, що формуються, мають можливість експортування в редактор електронних таблиць MS Office Excel.

### **5. Спеціальні вимоги**

Програма повинна мати дружній інтерфейс, розрахований на користувача середньої кваліфікації (з точки зору комп'ютерної грамотності). З огляду на обсяг проекту завдання передбачається вирішувати поетапно. При цьому модулі програмного забезпечення, створені в різний час, повинні передбачати можливість нарощування системи і бути сумісні один з одним; тому документація на прийняте експлуатаційне програмне забезпечення повинна містити повну інформацію, необхідну для роботи з ним програмістів. Мова програмування визначається вибором виконавця, при цьому він повинен забезпечувати можливість інтеграції програмного забезпечення з пакетом MS Office 2007/2010.

## 6. Вимоги до програмної документації

В ході розробки програми повинні бути підготовлені: текст програми, опис програми, програма і методика випробувань, керівництво користувача, техніко-економічне обґрунтування

## 7. Стадії та етапи розробки

Стадії та етапи розробки проекту Довідково-інформаційної системи «Ресторан» наведено в таблиці А.1.

Таблиця А.1 – Стадії та етапи розробки проекту

Стадія розробки	Етапи робіт	Зміст робіт
Технічне завдання 02.01.21 – 31.01.21	Обґрунтування необхідності розробки програми	Коротка характеристика програмного забезпечення; підстава і призначення розробки; вимоги до програмної системи і документація; стадії і етапи розробки програми; порядок контролю і приймання
Ескізний проект 01.02.21 – 14.02.21	Розробка ескізного проекту	Попередня розробка структури вхідних і вихідних даних; уточнення середовища програмування; розробка і опис загальної алгоритмічної структури системи, що буде розроблюватися
Технічний проект 15.02.21 – 28.02.21	Розробка технічного проекту	Уточнення структури вхідних і вихідних даних; розробка докладного алгоритму; розробка структури програми; остаточне визначення конфігурації технічних засобів
Робочий проект 01.03.21 – 10.04.21	Розробка програмного забезпечення	Реалізація програмного забезпечення; відладка; проведення попереднього тестування
Розробка програмної документації 11.04.21 – 20.04.21	Розробка документації до програмного забезпечення	Розробка необхідної документації, передбаченої технічним завданням
Тестування системи 21.04.21 – 30.04.21	Проведення тестування програмного забезпечення	Розробка методики тестування; проведення основних тестів; коректування програмного забезпечення
Впровадження	Підготовка і передача програми	Підготовка і передача програмного забезпечення; навчання персоналу використуванню програмного забезпечення; внесення коректувань в програмне забезпечення і документацію

## **8. Порядок контролю та приймання**

Контроль і приймання розробки здійснюються на основі розробленої методики випробувань. При цьому перевіряється виконання всіх функцій програми. Хід проведення випробувань документується у «Протоколі проведення випробувань». На основі цього протоколу підписується акт приймання-здачі програми в експлуатацію

ДОДАТОК Б  
(обов'язковий)

**ПРОГРАМНИЙ КОД**

```
<?php
class Page
{
    protected static $metaKeys = [
        'end',
        'first_page_uri',
        'next_page_uri',
        'last_page_uri',
        'page',
        'page_size',
        'previous_page_uri',
        'total',
        'num_pages',
        'start',
        'uri',
    ];

    protected $version;
    protected $payload;
    protected $solution;
    protected $records;

    abstract public function buildInstance(array $payload);

    public function __construct(Version $version, Response $response) {
        $payload = $this->processResponse($response);

        $this->version = $version;
        $this->payload = $payload;
        $this->solution = [];
        $this->records = new \ArrayIterator($this->loadPage());
    }

    protected function processResponse(Response $response) {
        if ($response->getStatusCode() !== 200 && !$this->isPagingEol($response->getContent())) {
            $message = '[HTTP ' . $response->getStatusCode() . '] Не знайдено';
            $code = $response->getStatusCode();

            $content = $response->getContent();
            $details = [];
            $moreInfo = "";
        }
    }
}
```

```

    if (\is_array($content)) {
        $message .= isset($content['message']) ? ' ' . $content['message'] : '';
        $code = $content['code'] ?? $code;
        $moreInfo = $content['more_info'] ?? '';
        $details = $content['details'] ?? [];
    }

    throw new RestException($message, $code, $response->getStatusCode(), $moreInfo,
    $details);
    }
    return $response->getContent();
    }

    protected function isPagingEol(?array $content): bool {
        return $content !== null && \array_key_exists('code', $content) && $content['code'] ===
20006;
    }

    protected function hasMeta(string $key): bool {
        return \array_key_exists('meta', $this->payload) && \array_key_exists($key, $this-
>payload['meta']);
    }

    protected function getMeta(string $key, string $default = null): ?string {
        return $this->hasMeta($key) ? $this->payload['meta'][$key] : $default;
    }

    protected function loadPage(): array {
        $key = $this->getMeta('key');
        if ($key) {
            return $this->payload[$key];
        }

        $keys = \array_keys($this->payload);
        $key = \array_diff($keys, self::$metaKeys);
        $key = \array_values($key);

        if (\count($key) === 1) {
            return $this->payload[$key[0]];
        }

        if ($this->isPagingEol($this->payload)) {
            return [];
        }

        throw new DeserializeException('Сторінка не може бути завантажена ');
    }

    public function getPreviousPageUrl(): ?string {

```

```

    if ($this->hasMeta('previous_page_url')) {
        return $this->getMeta('previous_page_url');
    } else if (\array_key_exists('previous_page_uri', $this->payload) && $this->payload['previous_page_uri']) {
        return $this->getVersion()->getDomain()->absoluteUrl($this->payload['previous_page_uri']);
    }
    return null;
}

public function getNextPageUrl(): ?string {
    if ($this->hasMeta('next_page_url')) {
        return $this->getMeta('next_page_url');
    } else if (\array_key_exists('next_page_uri', $this->payload) && $this->payload['next_page_uri']) {
        return $this->getVersion()->getDomain()->absoluteUrl($this->payload['next_page_uri']);
    }
    return null;
}

public function nextPage(): ?Page {
    if (!$this->getNextPageUrl()) {
        return null;
    }

    $response = $this->getVersion()->getDomain()->getClient()->request('GET', $this->getNextPageUrl());
    return new static($this->getVersion(), $response, $this->solution);
}

public function previousPage(): ?Page {
    if (!$this->getPreviousPageUrl()) {
        return null;
    }

    $response = $this->getVersion()->getDomain()->getClient()->request('GET', $this->getPreviousPageUrl());
    return new static($this->getVersion(), $response, $this->solution);
}

public function current() {
    return $this->buildInstance($this->records->current());
}

public function next(): void {
    $this->records->next();
}

```

```

public function key() {
    return $this->records->key();
}

public function valid(): bool {
    return $this->records->valid();
}

public function rewind(): void {
    $this->records->rewind();
}

public function getVersion(): Version {
    return $this->version;
}

public function __toString(): string {
    return '[Page]';
}
}

class AttemptModel
{
    const TYPE_FORGOT_PASS = 'FORGOT_PASS';
    const TYPE_SIGN_IN = 'SIGN_IN';
    const TYPE_EMAIL_VERIFY = 'TYPE_EMAIL_VERIFY';

    protected static $table = 'attempts';

    protected $queryKey = 'attempt_id';

    public function __construct()
    {
        parent::__construct();
    }

    public function logAttempt($type, $time = null, $userIP = null, $userID = 0)
    {
        if (!$time) {
            $time = time();
        }

        if (!$userIP) {
            $userIP = app()->request->getIp();
        }
    }
}

```

```

    }

    return $this->create([
        'attempt_type' => $type,
        'attempt_ip' => $userIP,
        'attempt_time' => $time,
        'user_id' => (int) $userID
    ]);
}

public function logSignInAttempt($time = null, $userIP = null)
{
    return $this->logAttempt(static::TYPE_SIGN_IN, $time, $userIP);
}

public function logForgotPassAttempt($time, $userIP = null, $userID = 0)
{
    return $this->logAttempt(static::TYPE_FORGOT_PASS, $time, $userIP, $userID);
}

public function logEmailVerificationAttempt($time, $userIP = null, $userID = 0)
{
    return $this->logAttempt(static::TYPE_EMAIL_VERIFY, $time, $userIP, $userID);
}

public function getLastAttempt(
    $type,
    $interval = null,
    $userIP = null,
    $userID = null,
    array $fields = ['*']
) {
    $sql = $this->select($fields)
        ->where('attempt_type', '=', $type)
        ->orderBy('attempt_time', 'DESC')
        ->limit(1, 0);

    if (is_string($userIP)) {
        $sql = $sql->where('attempt_ip', '=', $userIP);
    }

    if ($interval) {
        $sql = $sql->where('attempt_time', '>', $interval);
    }

    if (is_int($userID)) {

```

```

        $sql = $sql->where('user_id', '=', $userID);
    }

    $stmt = $sql->execute();
    return $stmt->fetch();
}

public function getAttemptCount($type, $interval, $userIP = null, $userID = null)
{
    $sql = $this->select(["COUNT(*) AS total"])
        ->where('attempt_type', '=', $type)
        ->where('attempt_time', '>', $interval);

    if (is_int($userID)) {
        $sql = $sql->where('user_id', '=', $userID);
    }

    if (is_string($userIP)) {
        $sql = $sql->where('attempt_ip', '=', $userIP);
    }

    $stmt = $sql->execute();
    $count = $stmt->fetch();

    return (int) $count['total'];
}

public function getForgotPassAttemptCount($interval, $userIP = null, $userID = null)
{
    return $this->getAttemptCount(static::TYPE_FORGOT_PASS, $interval, $userIP, $userID);
}

public function getSignInAttemptCount($interval, $userIP = null, $userID = null)
{
    return $this->getAttemptCount(static::TYPE_SIGN_IN, $interval, $userIP, $userID);
}

public function getEmailVerificationAttemptCount($interval, $userIP = null, $userID = null)
{
    return $this->getAttemptCount(static::TYPE_EMAIL_VERIFY, $interval, $userIP, $userID);
}
}

class RoleModel

```

```

{

const TYPE_ADMIN = 1;

const TYPE_MOD  = 2;

const TYPE_USER = 3;

protected $sortRules = [
    'protected' => ['is_protected' => 'DESC'],
    'custom' => ['is_protected' => 'ASC'],
    'newest' => ['created_at' => 'DESC'],
    'oldest' => ['created_at' => 'ASC'],
    'a2z' => ['role_name' => 'ASC'],
    'z2a' => ['role_name' => 'DESC'],
];

protected $autoTimestamp = true;

protected static $table = 'roles';

protected $queryKey = 'role_id';

protected $rolePerms = [];

protected $permissions;

public function getAllPermissions()
{
    if (is_array($this->permissions)) {
        return $this->permissions;
    }

    $db = app()->db;
    $table = Model::getPrefix('permissions');
    $sql = $db->select()->from($table);
    $stmt = $sql->execute();
    $perms = [];

    foreach ($stmt->fetchAll() as $row) {

```

```

        $perms[$row['perm_id']] = $row['perm_desc'];
    }

    $this->permissions = $perms;

    return $this->permissions;
}

```

```

public function addPermission($name)
{
    $permID = $this->getPermission($name);

    if ((int) $permID) {
        return $permID;
    }

    $db = app()->db;
    $table = Model::getPrefix('permissions');
    return $db->insert(['perm_desc' => $name])
        ->into($table)
        ->execute();
}

```

```

public function getPermission($name)
{
    $db = app()->db;
    $table = Model::getPrefix('permissions');
    $perm = $db->select(['perm_id'])
        ->from($table)
        ->where('perm_desc', '=', $name)
        ->execute()
        ->fetch();

    if (isset($perm['perm_id'])) {
        return $perm['perm_id'];
    }

    return false;
}

```

```

public function getRolePerms($roleID)
{
    if (isset($this->rolePerms[$roleID])) {
        return $this->rolePerms[$roleID];
    }

    $table1 = Model::getPrefix('role_perm');

```

```

$stable2 = Model::getPrefix('permissions');

$sql = "SELECT t2.perm_desc, t2.perm_id FROM {$stable1} as t1
      JOIN {$stable2} as t2 ON t1.perm_id = t2.perm_id
      WHERE t1.role_id = :role_id";
$sth = app()->db->prepare($sql);
$sth->execute(["role_id" => $roleID]);

$perms = [];

while ($row = $sth->fetch()) {
    $perms[$row["perm_desc"]] = $row['perm_id'];
}

$this->rolePerms[$roleID] = $perms;

return $perms;
}

public function insertPerm($roleID, $permID)
{
    $db = app()->db;
    $stable = Model::getPrefix('role_perm');

    $sql = $db->insert(['role_id', 'perm_id']->into($stable)->values([$roleID, $permID]);
    $sql->execute();

    return true;
}

public function deletePerm($roleID, $permID)
{
    $db = app()->db;
    $stable = Model::getPrefix('role_perm');
    return $db->delete()
        ->from($stable)
        ->where('role_id', '=', $roleID)
        ->where('perm_id', '=', $permID)
        ->execute();
}

public function deleteRolePerms($roleID)
{
    $db = app()->db;
    $stable = Model::getPrefix('role_perm');
    return (bool) $db->delete()
        ->from($stable)

```

```

        ->where('role_id', '=', $roleID)
        ->execute();
    }

    public function deleteRole($roleID)
    {
        $db = app()->db;
        $table1 = Model::getPrefix('roles');
        $table2 = Model::getPrefix('role_perm');

        $sql = "DELETE t1, t2 FROM {$table1} as t1
        JOIN {$table2} as t2 on t1.role_id = t2.role_id
        WHERE t1.role_id = :role_id";

        $sql = $db->prepare($sql);
        $sql->bindParam(":role_id", $roleID, \PDO::PARAM_INT);

        return $sql->execute();
    }

    public function getUsersCountUnderRole($roleID)
    {
        static $userModel;
        if (!$userModel) {
            $userModel = new UserModel;
        }

        $filters = [];
        $filters['where'][] = [$this->queryKey, '=', $roleID];

        return $userModel->countRows(null, $filters);
    }
}

class ContentModel
{
    const TYPE_PAGE = 'page';
    const TYPE_ATTACHMENT = 'attachment';
    const TYPE_POST = 'post';

    protected static $table = 'content';

    protected $queryKey = 'content_id';

    protected $autoTimestamp = true;

```

```
protected $sortRules = [
    'newest'    => ['created_at' => 'DESC'],
    'oldest'    => ['created_at' => 'ASC'],
    'a2z'       => ['content_title' => 'ASC'],
    'z2a'       => ['content_title' => 'DESC'],
];
```

```
public function addAttachment($path, array $data)
{
    $data['content_type'] = 'attachment';
    $data['content_path'] = $path;
    $data['content_body'] = "";

    if (isset($data['content_meta']) && is_array($data['content_meta'])) {
        $data['content_meta'] = json_encode($data['content_meta']);
    } else {
        unset($data['content_meta']);
    }

    if (empty($data['content_author'])) {
        $data['content_author'] = current_user_ID();
    }

    if (isset($data['content_slug'])) {
        unset($data['content_slug']);
    }

    return $this->create($data);
}
```

```
public function deleteAttachment($contentID)
{
    $filters = [];
    $filters['where'][] = ['content_type', '=', 'attachment'];
    $attachment = $this->read($contentID, ['content_path'], $filters);
    if (!$attachment) {
        return false;
    }

    $this->delete($contentID);

    $file = uploadspath($attachment['content_path']);
    return @unlink($file)
}
```

```

public static function addAllowedFileTypes(array $fileTypes)
{
    static::$allowedFileTypes = array_merge(static::$allowedFileTypes, $fileTypes);
    return true;
}

```

```

public static function getAllowedFileTypes()
{
    return static::$allowedFileTypes;
}

```

```

public static function getFileType($ext, $mimetype)
{
    if (preg_match('#spreadsheet|excel#i', $mimetype)) {
        return 'spreadsheet';
    }
    if (preg_match('#.presentation|powerpoint#i', $mimetype)) {
        return 'interactive';
    }
    if (preg_match('#officedocument#i', $mimetype)) {
        return 'document';
    }
    if (preg_match('#^text/#i', $mimetype)) {
        return 'text';
    }

    if (preg_match('#^image/#i', $mimetype)) {
        return 'image';
    }

    if (preg_match('#^video/#i', $mimetype)) {
        return 'video';
    }

    if (preg_match('#^audio/#i', $mimetype)) {
        return 'audio';
    }

    $ext = mb_strtolower($ext);
}
}

```

```

public function isValidSortMimeType($type)

```

```
{  
    return in_array(mb_strtolower($type), $this->mimeSortType);  
}  
}
```

ДОДАТОК В  
(обов'язковий)

**ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ**

Хмельницький національний університет  
Факультет програмування та комп'ютерних і телекомунікаційних  
систем Кафедра інженерії програмного забезпечення

Тема дипломного проекту

Довідково-інформаційна система «Ресторан»

Студента IV курсу, група ПЗ-17-1 Красюк Яни

Керівник ст. викладач, Г.І. Бедратюк

*Мета роботи* – розробка довідково-інформаційної системи для автоматизації роботи ресторану.

*Практична значимість* полягає в можливості використання спроектованого сайту для інформаційної підтримки діяльності ресторану, в першу чергу, для завоювання лояльності відвідувачів ресторану.

Однією з перспективних галузей, що розвиваються, є ресторанний бізнес. Досягнення успіху у розвитку будь-якого бізнесу, зокрема ресторанного, залежить від багатьох факторів. Процес управління ресторанним бізнесом є доволі складним та вимагає: контролю за обліковим процесом та поведінкою персоналу, аналізу транзакцій, обліку надходження продуктів, формування вартості страв і напівфабрикатів, процедур списання продуктів, дотримання санітарних і технологічних норм.

Потреба автоматизації всіх цих процесів впливає з необхідності врахування великої кількості деталей.

Зручність автоматизації процесів у закладах громадського харчування очевидна не тільки з погляду власника, але й з позиції клієнтів, оскільки довідково-інформаційні системи дають змогу виконувати розрахунки з відвідувачами оперативніше, забезпечити систему знижок для зареєстрованих клієнтів, здійснювати обслуговування на основі черги, забезпечити меню усіма необхідними інгредієнтами та заощадити час.

Основні  
виділені  
характеристики  
сайту

№ п/п	Характеристика
<b>Інформація</b>	
1.	Наявність інформації про функції, властивості, ціни товару
2.	Наявність корпоративної інформації (фінансові звіти, вакансії, мета)
3.	Наявність контактної інформації (телефон, адреса)
4.	Наявність інформації про тенденції галузі, галузеві огляди
5.	Наявність розділу «Питання які найчасіше задають»
6.	Наявність відгуків про компанію і товари
7.	Історія компанії

Основні  
виділені  
характеристики  
сайту

№ п/п	Характеристика
<b>Привабливість</b>	
8.	Наявність фото товару та відео про товар
9.	Оформлення і привабливість сайту
10.	Фотографії клієнтів
11.	Фотографії персоналу компанії та її керівника

Основні  
виділені  
характеристики  
сайту

№ п/п	Характеристика
<b>Простота</b>	
12.	Простота навігації
13.	Ясність та зрозумілість тексту

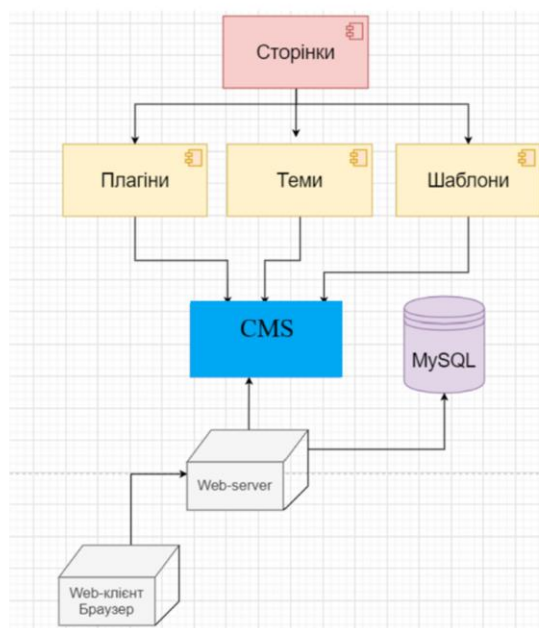
Основні  
виділені  
характеристики  
сайту

№ п/п	Характеристика
<b>Додаткові можливості</b>	
14.	Наявність інструкцій по використанню, поради по підбору моделей, кольору, розміру
15.	Можливість збереження персональних налаштувань
16.	Наявність реклами інших компаній
17.	Наявність онлайн гри та інших розваг
18.	Можливість безкоштовного скачування додаткових програм

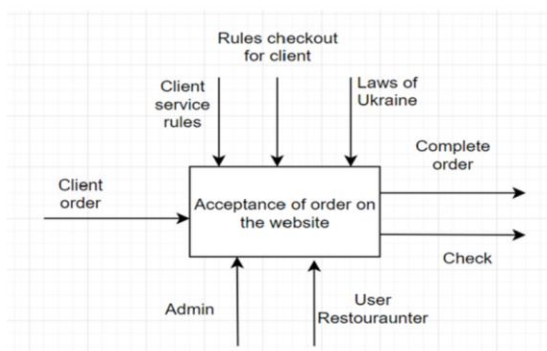
## Основні виділені характеристики сайту

№ п/п	Характеристика
<b>Зворотній зв'язок</b>	
19.	Можливість здійснити онлайн-покупку
20.	Онлайн-діагностика виникаючих проблем при використанні товару
21.	Можливість відстежувати статус замовлення і його місце знаходження

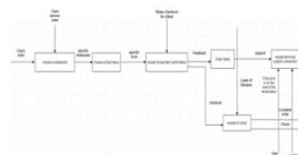
Дана довідково-інформаційна система містить такі компоненти як плагіни, шаблони, теми, сторінки та базу даних. Саму архітектуру довідково-інформаційної системи можна розглянути на рис.

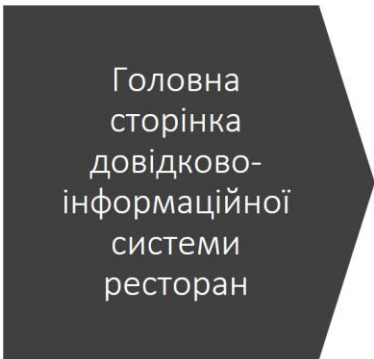
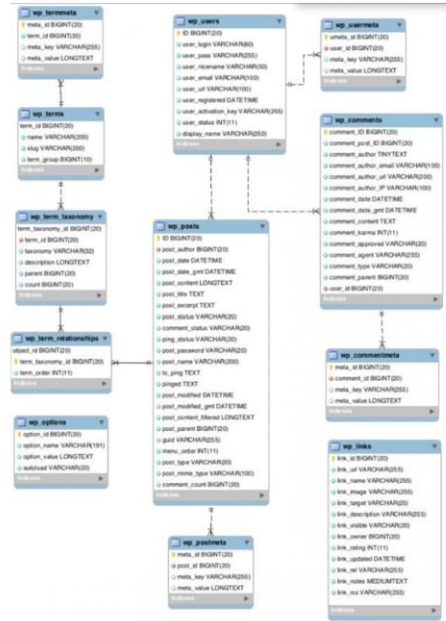
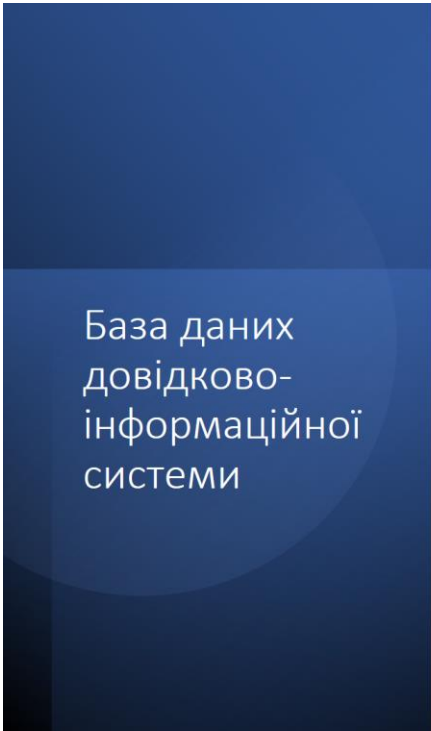


Початком розробки проекту можна вважати створення контекстної діаграми. Вона містить короткі, місткі, лаконічні дані які показують особливості взаємодії довідково-інформаційної системи та оточення і описують їх зміст.



Основна властивість контекстної діаграми полягає в зображенні основної суті веб-додатку, що потребує необхідності створення декомпозиції. Декомпозиція надає дані про особливості роботи веб-додатку.  
Принципи декомпозиції:  
Клієнтське замовлення  
Обрання страви з меню  
Звіт по оплаті: зроблена оплата чи ні. Якщо оплата відбулася – то видача чеку, якщо ні то звернення до служби підтримки





**РЕСТОРАН**

### Страви дня

Кожен день ми створюємо особливу страву для гурманів



**Бутерброди**

Паштети з риби або риби консерви - наприклад, горбуша або тунець, чудово підходять для приготування шведських і австрійських закусок. Трохи зелені цибули, сиру або майонезу, кавари зробить смак бутерброда яскравіше ...



**Салати**

Коли згадується слово «салат», відразу ж хочеться додати: «Оливки», «Шуба», «Грибний», «Рошшика», «Кривавий» (у кого, що ближче до серця лежить). Насправді ж з італійської salata - «солонка (зелень)», покріне від ...




**Брускета**

традиційна італійська закуска. Відмінна риса брускети від бутерброда чи тоста - в тому, що сирихача хліба попередньо обсмажують на грилі чи на сковорідці без олії. Ідеально для приготування брускети підходить ...


Де є  
можливість  
переглянути  
все меню

Кожен день ми створюємо особливу страву для гурмана




**Бутерброди**

Паштети з риби або рибні консерви - наприклад, горбуша або тунець, чудово підходять для приготування швидких і апетитних закусок. Трохи зеленої цибули, сиру або майонезу, кав'яр зроблять смак бутербродів яскравіше ...



**Салати**

Коли згадується слово «салат», відразу ж хочеться додати «Оліє»-«Шуба», «Грецький», «Ромашка», «Крабова» (у кого, що ближче до серця лежить), «Насправді» ж з італійської salata - «солонка (зелена)», покладіть від ...



**Брускетта**

традиційна італійська закуска. Відмінна риса брускетти від бутерброда чи тоста - в тому, що сиричкові хліба попередньо обсмажують на грилі чи на сковорідці без олії. Ідеально для приготування брускетти підходять ...

[ПЕРЕГЛЯНУТИ ВСЕ МЕНЮ →](#)

З інформацією  
про місце  
розташування

**Ми пропонуємо**

Друзі! Навіть карантин в не завадить вам насолодитися улюбленими стравами! Відтепер наш заклад працюватиме з 10 по 22 на вніс за спеціальним меню!

- Відкриті**  
Замовити їжу через вікно торгівлі або зайти до закладу та замовити страву на вніс.
- Вільний паркінг**  
Ресторан у центрі міста з вільним паркінгом.
- Розташування в центрі міста**  
Ми розташовані в історичній частині міста, за 5 хвилин ходьби від площі незалежності



## Переглядом меню

### Наше меню

М'ясо, рибе, вегетаріанська, карта вин, карта коктейлів, десерти.

Салати   Основні страви   Десерти   Перші страви

**Сирний крем суп** #45,00

Ніжна текстура сирного крем-супу заховає вас з першої ложки.

[Додати у кошик](#)

**Салат Рікота** #140,00

Свіжі овочі: морква, огірок, селера, солодкий перець, з сиром фета у ніжному соусі Рікота

[Додати у кошик](#)

**Салат від шеф-кухара** #270,00

Ніжна, із злегка тертим горіховим смаком, приємно-зелений м'якуш авокадо надасть свіжому несповторній, а м'ясо тунця одрим відшуканого ніжного салазу не лише є легким поживним продуктом, але і цінується за насиченість білком і гемоглобіном.

[Додати у кошик](#)




**Класичний салат Кобб** #250,00

салат Кобб готується з обсмаженого бекону і курячого філе, заправки круто їєдь – це та частина інгредієнтів, яка вимагає терасорборбих. Перепелині яйця можна замінити на пару курячих, овочева частина – не виключе складноціа.

[Додати у кошик](#)

## Можливістю оплати

### Кошик

	Товар	Ціна	Кількість	Проміжний підсумок
×	 Салат Рікота	#140,00	<input type="text" value="2"/>	#280,00
×	 Класичний салат Кобб	#250,00	<input type="text" value="2"/>	#500,00
×	 Зелений салат	#80,00	<input type="text" value="1"/>	#80,00

Код купону  [ЗАСТОСУВАТИ КУПОН](#) [ОНОВИТИ КОШИК](#)

#### Підсумки кошика

Проміжний підсумок	#660,00
Загалом	#860,00

[ПЕРЕЙТИ ДО ОФОРМЛЕННЯ](#)

## ВИСНОВОК

У даній роботі було зроблено дослідження предметної області із виділенням функціональних та нефункціональних вимог до розроблюваного програмного забезпечення та окреслене технічне завдання і зроблена постановка задачі. Також було проаналізовано існуючі інформаційно-довідкові системи та виявлено плюси та мінуси наявного програмного забезпечення.

Завідувачу кафедри інженерії програмного забезпечення проф. Бедратюку Л. П.

здобувача вищої освіти

Красюк. Я. І.

Прізвище, ініціали

факультет ПКТС, 4 курс, група ПЗ-17-1

### ЗАЯВА

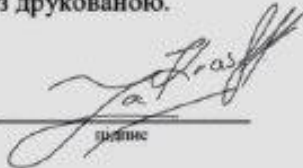
З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення 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 плагіату в текстах робіт.

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

1.06.2021

дата

  
\_\_\_\_\_

підпис

**Anti-Plagiarism v-15.257****Максимальне співпадіння з одним документом 4.0%****Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 12%**

ID: 93540 Назва: Довідково-інформаційна система «Ресторан» Додано в БД: 2021-06-14 Автора: Я. І. Красюк Керівники: Г. І. Бедратюк Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	78812	1184	6125 (8%)	98 (8%)

## Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми



Ім'я користувача:  
Кафедра ІПЗ

ID перевірки:  
1008287907

Дата перевірки:  
14.06.2021 10:43:36 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
14.06.2021 10:45:46 EEST

ID користувача:  
100005589

Назва документа: **Дипломний проект Красюк Я.І. без додатків**

Кількість сторінок: 67 Кількість слів: 11540 Кількість символів: 90109 Розмір файлу: 2.59 MB ID файлу: 1008357163

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

**20.9%**  
**Схожість**

Найбільша схожість: 6.34% з Інтернет-джерелом (<http://librus.dobrota.biz/40yuridicheskie/262302-1-otdelenie-shkoli-noc>)

13.7% Джерела з Інтернету

278

Сторінка 69

8.27% Джерела з Бібліотеки

87

Сторінка 71

**0% Цитат**

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

**0%**  
**Вилучень**

Немає вилучених джерел

**Модифікації**

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

18

Підозріле форматування

12  
сторінок

## ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНИЙ ПРОЕКТ  
освітнього ступеня «Бакалавр»Дипломник Красюк Яна ІванівнаТема Довідково-інформаційна система «Ресторан»Спеціальність 121 – Інженерія програмного забезпечення**Обсяг дипломного проекту:**Кількість листів креслень \_\_\_\_\_; кількість сторінок записки 14

1. Короткий зміст пояснювальної записки та прийнятих рішень. У дипломному проекті було досліджено предметну область довідково-інформаційної системи та усі її функціональні та нефункціональні вимоги. Був проведений аналіз існуючих програм, розглянуто їх переваги і недоліки, та доведено актуальність розробки нового програмного забезпечення.

2. Висновок про відповідність проекту поставленому завданню. Дипломний проект виконаний відповідно до поставленого завдання та з дотриманням усіх вимог.

3. Характеристика виконання кожного розділу проекту, ступінь використання останніх досягнень науки і техніки та передових методів роботи. У першому розділі було проведено дослідження предметної області та виділення функціональних та нефункціональних вимог до довідково-інформаційної системи та здійснено постановку задачі. У другому розділі проведено проектування, а саме архітектуру даної системи. Розроблено структуру та спроектовано базу даних та інтерфейс користувача. Далі була розроблена база даних та модулів системи. Розроблена інструкція користувача, також проводилося налагодження та тестування системи.

4. Позитивні сторони проекту. Дана довідково-інформаційна система розроблена для оптимізації роботи закладу громадського харчування, його популяризації в мережі Інтернет, та покращенню зв'язку між клієнтами та рестораном. І також вона є зручна у використанні.

5. Негативні сторони проекту. У даній роботі потрібно доповнити інформаційно-довідкову систему «Ресторан» ще більшою інформацією і картинками меню, для ще більш зручного і привабливого користування сайтом.

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

7. Відгук про дипломний проект в цілому Дипломний проект є структурований, послідовний і чіткий та простий, що дозволяє чітко зрозуміти викладений матеріал у рамках тематики дипломного проекту. Графічний матеріал дає можливість наочно побачити деталі проектування системи, також він є зручним у використанні.

8. Інші зауваження \_\_\_\_\_

9. Оцінка дипломного проекту Дипломний проект виконаний у повному обсязі і відповідає поставленій задачі та заслуговує на оцінку «добре».

РЕЦЕНЗЕНТ Кисіль Тетяна Миколаївна кандидат фіз.-мат. наук, доцент кафедри комп'ютерної інженерії та системного програмування (КІСП) ХНУ

«15» серпня

2021 р.

  
(підпис)

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

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: «Довідково-інформаційна система Ресторан»

Автор: Красюк Яна Іванівна

Спеціальність: 121 – Інженерія програмного забезпечення

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

Науковий керівник: Бедратюк Ганна Іванівна, ст. викладач

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є 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. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але 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 досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби ухриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

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

- 1) у тексті дипломного проекту системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень в бланках (титулка, бланк завдання, в структурі підрозділів ВСТУПУ) та в назвах публікацій джерел;
- 2) В якості запозичень системою було зафіксовано послідовність вихідного коду, які є спільними для великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
- 3) усі запозичення фрагментарні, або мають належним чином оформленні посилання.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності / схожості, складає 20,9% і адресується до 365 періоджерел, що з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь дипломного проекту.

Керівник



Г.І. Бедратюк

Гарант ОП



Л.П. Бедратюк

Завідувач кафедри



Л.П. Бедратюк