

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:

«Клієнт-серверний додаток ресторану італійської кухні»

Студента 4 курсу, 7 групи,
спеціальності 121 «Інженерія
програмного забезпечення»
освітньої програми «Інженерія
програмного забезпечення»

Вишненка Володимира
Володимировича

підпис студента

Науковий керівник
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

Рзаєва Світлана
Леонідівна

підпис керівника

Гарант освітньої програми
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

Рзаєва Світлана
Леонідівна

підпис гаранта

Державний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

«14» листопада 2022 р.

Завдання

на випускний кваліфікаційний проєкт студентіві

Вишненку Володимиру Володимировичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Клієнт-серверний додаток
ресторану італійської кухні»

Затверджена наказом ректора від «6» грудня 2022 р. № 3288

2. Строк здачі студентом закінченого проєкту 5 червня 2023

3. Цільова установка та вихідні дані до проєкту

Мета проєкту розробка клієнт-серверного додатку для ресторану італійської
кухні з метою поліпшення процесів управління рестораном, зокрема
бронювання столиків та забезпечення зручності для клієнтів.

Об'єкт дослідження клієнт-серверний додаток ресторану італійської кухні.

Предмет дослідження технології розробки клієнт-серверного додатку
ресторану італійської кухні.

4. Консультанти проєкту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проєкту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1 ОСОБЛИВЛІСТІ РОЗРОБКИ КЛІЄНТ-СЕРВЕРНОГО ДОДАТКУ РЕСТОРАНУ ІТАЛІЙСЬКОЇ КУХНІ

- 1.1. Властиві риси італійської кухні
- 1.2. Клієнт-серверна модель додатку
- 1.3. Загальна структура сайту
- 1.4. Технічне завдання на програмне забезпечення
- 1.5. Висновки до розділу 1

РОЗДІЛ 2 ПРОГРАМНІ ЗАСОБИ ПРОЄКТУВАННЯ КЛІЄНТ-СЕРВЕРНОГО ДОДАТКУ

- 2.1. Опис архітектури додатку
- 2.2. Моделі та діаграми: для чого і коли створюються
- 2.3. Еталонна модель сайту ресторану
- 2.3. Діаграма стану бронювання столика в ресторані
- 2.4. Діаграма класів сайту ресторану
- 2.5. Висновки до розділу 2

РОЗДІЛ 3 РЕАЛІЗАЦІЯ КЛІЄНТ-СЕРВЕРНОГО ДОДАТКУ

- 3.1. Вибір програмного забезпечення для побудови та налагодження бази даних та її структури
- 3.2. Використання css, html та javascript
- 3.3. Розробка форми для бронювання столиків
- 3.4. Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання проєкту

№ пор.	Назва етапів випускного кваліфікаційного проєкту	Строк виконання етапів проєкту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проєкту</i>	21.09.2022	21.09.2022
2.	<i>Розробка та затвердження завдання на проєкт</i>	14.11.2022	14.11.2022
3.	<i>Вступ та перелік літературних джерел</i>	23.12.2022	23.12.2022
4.	<i>Розділ 1. Особливості розробки клієнт-серверного додатку ресторану італійської кухні</i>	27.01.2023	27.01.2023
5.	<i>Розділ 2. Програмні засоби проєктування клієнт-серверного додатку</i>	03.03.2023	03.03.2023
6.	<i>Розділ 3. Реалізація клієнт-серверного додатку</i>	14.04.2023	14.04.2023
7.	<i>Висновки</i>	28.04.2023	28.04.2023
8.	<i>Задача випускного кваліфікаційного проєкту на кафедрі (перша перевірка)</i>	17.05.2023	17.05.2023
9.	<i>Підготовка автореферату та презентації доповіді</i>	26.05.2023	26.05.2023
10.	<i>Попередній захист випускного кваліфікаційного проєкту</i>	29.05.2023 – 02.06.2023	
11.	<i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>	05.06.2023	05.06.2023
12.	<i>Задача прошитого випускного кваліфікаційного проєкту на кафедрі</i>	05.06.2023	05.06.2023
13.	<i>Публічний захист випускного кваліфікаційного проєкту</i>		

7. Дата видачі завдання «14» листопада 2022 р.

8. Науковий керівник випускного кваліфікаційного проєкту _____

Рзаєва С.Л.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми _____

Рзаєва С.Л.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент _____

Вишненко В.В.

(прізвище, ініціали, підпис)

АНОТАЦІЯ

Випускний кваліфікаційний проєкт присвячений розробці та реалізації сайту ресторану з функціоналом бронювання столиків. Метою роботи є створення зручної та ефективної онлайн-платформи, що дозволить клієнтам легко та швидко бронювати столики в ресторані.

У роботі розглянуті та використані новітні підходи і технології, такі як веб-розробка, база даних і серверні технології. Для зручності користувачів буде розроблено інтуїтивно зрозумілий інтерфейс для перевірки наявності місць, вказівки дати і часу, вказівки кількості осіб і бронювання.

Клієнтська частина була розроблена з використанням мови програмування JavaScript, мови розмітки HTML і мови стилів CSS.

Ключові слова: ресторан, італійська кухня, клієнт-серверний додаток, база даних, веб-розробка, JavaScript, замовлення, резервація, меню, інтерфейс, користувач, ефективність, автоматизація, управління рестораном.

ABSTRACT

The final qualification project is dedicated to the development and implementation of a restaurant website with table reservation functionality. The aim of the work is to create a convenient and efficient online platform that will allow customers to book tables in a restaurant easily and quickly.

The latest approaches and technologies, such as web development, database and server technologies, will be studied and used in the work. For the convenience of users, an intuitive interface will be developed to check availability, specify the date and time, indicate the number of people and make reservations.

The client side was developed using the JavaScript programming language, HTML markup language and CSS style language.

Keywords: restaurant, Italian cuisine, client-server application, database, web development, JavaScript, order, reservation, menu, interface, user, efficiency, automation, restaurant management.



ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1 ОСОБЛИВЛІСТІ РОЗРОБКИ КЛІЄНТ-СЕРВЕРНОГО ДОДАТКУ РЕСТОРАНУ ІТАЛІЙСЬКОЇ КУХНІ.....	5
1.1. Властиві риси італійської кухні.....	5
1.2. Клієнт-серверна модель додатку.....	7
1.3. Загальна структура сайту.....	9
1.4. Технічне завдання на програмне забезпечення.....	12
1.5. Висновки до розділу 1.....	15
РОЗДІЛ 2 ПРОГРАМНІ ЗАСОБИ ПРОЄКТУВАННЯ КЛІЄНТ-СЕРВЕРНОГО ДОДАТКУ.....	16
2.1. Опис архітектури додатку.....	16
2.2. Моделі та діаграми: для чого і коли створюються.....	18
2.3. Еталонна модель сайту ресторану.....	20
2.3. Діаграма стану бронювання столика в ресторані.....	21
2.4. Діаграма класів сайту ресторану.....	24
2.5. Висновки до розділу 2.....	25
РОЗДІЛ 3 РЕАЛІЗАЦІЯ КЛІЄНТ-СЕРВЕРНОГО ДОДАТКУ.....	27
3.1. Вибір програмного забезпечення для побудови та налагодження бази даних та її структури.....	27
3.2. Використання CSS, HTML та JavaScript.....	30
3.3. Розробка форми для бронювання столиків.....	32
3.4. Висновок до розділу 3.....	33
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	35
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	36
ДОДАТКИ.....	

					<i>ДТЕУ 121 07-04.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Клієнт-серверний додаток ресторану італійської кухні	<i>Стадія</i>	<i>Арку</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		23.12.22		3	2	36
Керівник		Рзаєва С.Л.		23.12.22		Факультет інформаційних технологій 4 курс, 7 група		
Гарант		Рзаєва С.Л.		23.12.22				
Розробив		Вишненко В.В.		23.12.22	<i>Зміст</i>			

ВСТУП

Актуальність. З ростом чисельності ресторанів стрімко посилюється і конкуренція, що неминуче призводить до необхідності ефективно і раціонально використовувати наявні ресурси. У цих умовах для успішного ведення бізнесу необхідно інвестувати в засоби та інструменти його підтримки та розвитку. Один з основних інструментів розвитку ресторанного бізнесу – це автоматизація роботи ресторану. Італійська кухня є дуже популярною у світі, тому ресторани, які пропонують цей вид кулінарних страв, привертають все більше клієнтів. Розробка клієнт-серверного додатку для такого ресторану допоможе поліпшити процес обслуговування гостей та підвищити продуктивність закладу. Розробка зручного та ефективного способу бронювання столиків та забезпечення зворотного зв'язку з клієнтами допоможе поліпшити взаємодію з ними та підвищити рівень задоволеності клієнтів від обслуговування.

Автоматизація ресторанного бізнесу через розробку клієнт-серверного додатку є актуальним завданням, оскільки вона сприяє ефективному веденню ресторанної діяльності і поліпшенню якості обслуговування. Зокрема, забезпечення зручного бронювання столиків за допомогою додатку дозволяє клієнтам легко і швидко здійснити необхідні дії, підвищуючи їх задоволеність. Розроблений додаток, використовуючий потужність інтернет-технологій, має потенціал стати ефективним інструментом управління рестораном і відповідати сучасним вимогам галузі.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 07-04.БР</i>			
Зав. каф.		Криворучко О.В.		23.12.22	Клієнт-серверний додаток ресторану італійської кухні	Стадія	Аркуш	Аркушів
Керівник		Рзаєва С.Л.		23.12.22		В	3	36
Гарант		Рзаєва С.Л.		23.12.22		Факультет інформаційних технологій		
Розробив		Вишненко В.В.		23.12.22		4 курс, 7 група		
					<i>Вступ</i>			

Мета дослідження: розробка клієнт-серверного додатку для ресторану італійської кухні з метою поліпшення процесів управління рестораном, зокрема бронювання столиків та забезпечення зручності для клієнтів.

Об'єкт дослідження: клієнт-серверний додаток ресторану італійської кухні.

Предмет дослідження: технології розробки клієнт-серверного додатку ресторану італійської кухні.

У відповідності з метою дослідження поставлені наступні завдання:

- розробити архітектуру та функціональні вимоги до клієнт-серверного додатку ресторану італійської кухні;
- описати технічне завдання програмного додатку;
- розробити логічну та фізичну моделі бази даних програмного додатку ресторану італійської кухні;
- використовуючи мову програмування JavaScript, реалізувати серверну частину додатку, включаючи базу даних та логіку обробки запитів;
- розробити клієнтську частину додатку, яка буде забезпечувати зручний та інтуїтивно зрозумілий інтерфейс для користувачів.

Практичне значення дослідження: полягає у впровадженні розробленого клієнт-серверного додатку для ресторану італійської кухні, що дозволить поліпшити процес обслуговування гостей, бронювання столиків. Це сприятиме підвищенню задоволення клієнтів, залученню нових гостей, покращенню репутації ресторану і збільшенню його ефективності та прибутковості.

						ДТЕУ 121 07-04.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			4

РОЗДІЛ 1

ОСОБЛИВЛІСТІ РОЗРОБКИ КЛІЄНТ-СЕРВЕРНОГО ДОДАТКУ РЕСТОРАНУ ІТАЛІЙСЬКОЇ КУХНІ

1.1. Властиві риси італійської кухні

Італійська їжа набагато простіша, ніж, наприклад, французька, і не має тієї розкішності, яка асоціюється з кухнею інших європейських країн, але їй властива приваблива легкість настрою і кулінарного ставлення. Вона складається з поєднання зернових, овочів, риби, фруктів, м'яса і сиру, які вживаються з оливковою олією. В Італії теплий субтропічний клімат, вирощується багато овочів і фруктів, особливо в південних регіонах країни, що має значний вплив на харчові вподобання мешканців. Капуста, артишоки, квасоля, спаржа та цибуля вважаються основними овочами італійської кухні. У традиційній італійській кухні овочі не варяться і не смажаться, а тушкуються у власному соку з додаванням оливкової олії або вина. На півночі є багато рівнинних пасовищ, де вирощують велику рогату худобу. Там виробляють багато м'яса, молока і сиру. Як це часто буває, такий поділ країни на регіони з різними смаковими традиціями позитивно впливає на національний раціон в цілому.

Італійська кухня також відома своїми стравами, характерними для конкретного регіону, що необхідно враховувати під час розробки програми. Наприклад, у деяких регіонах Італії є страви та рецепти, характерні тільки для їхнього регіону, що необхідно врахувати в меню застосунку. Додаток також може надавати інформацію про походження страви, історію її приготування та інформацію, що відображає культурне тло італійської кухні.

Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121 07-04.БР			
Зав. каф.		Криворучко О.В.		27.01.23	Клієнт-серверний додаток ресторану італійської кухні	Стадія	Аркуш	Аркушів
Керівник		Рзаєва С.Л.		27.01.23		РІ	5	36
Гарант		Рзаєва С.Л.		27.01.23		Факультет інформаційних технологій 4 курс, 7 група		
Розробив		Вишненко В.В.		27.01.23				
					Особливості розробки клієнт-серверного додатку ресторану італійської кухні			

Це дасть змогу не тільки задовольнити вподобання ваших клієнтів, а й розширити їхні знання про італійську культуру та традиції. Взавши до уваги особливості італійської кухні та розробивши клієнт-серверний застосунок, ви зможете надати своїм клієнтам автентичний і задовільний ресторанный досвід, забезпечивши своєму закладу конкурентну перевагу та успішну роботу.

В італійській кухні представлена велика різноманітність видів пасти, від легких і тонких до товстих і багат шарових за текстурою. Існує безліч відомих видів пасти, включно зі спагеті, феттучіне, лазаньєю, равіолі й тальятелле. У різних регіонах використовуються різні види пасти і соусів, що робить італійську кухню дуже різноманітною і цікавою для гастрономів. Оливкова олія є одним з основних інгредієнтів італійської кухні та відіграє важливу роль у формуванні смаку й аромату багатьох страв. Італійські оливки також часто з'являються на обідньому столі і використовуються як інгредієнт в оливковому песто і соусах. Країна також славиться своїми сирами, включно з моцарелою, пармезаном і горгонзолою. Ці сири є невід'ємним елементом багатьох італійських страв, надаючи їм особливого смакового акценту. Італійська кава, особливо еспресо, також посідає важливе місце в італійській культурі харчування. Каву створюють майстри-бариста, які ретельно відбирають зерна і приділяють увагу кожній деталі процесу приготування. Італійський спосіб приготування кави є джерелом натхнення для багатьох кав'ярень по всьому світу. Морепродукти популярні на півдні Італії, особливо на Сицилії та в Неаполі, де можна знайти найрізноманітніші гастрономічні шедеври. Кальмари, мідії, тріска й омари є основою для багатьох страв із морепродуктів. Водночас італійська кухня відома своєю філософією повільного харчування, в якій особлива увага приділяється якості, свіжості та натуральності продуктів, а також невимушеній трапезі в приємній атмосфері.

						ДТЕУ 121 07-04.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			6

Загалом розуміння унікальності та регіональних впливів італійської кухні дасть вам змогу розробити докладніший та комплексніший додаток, що відповідатиме уподобанням і пізнавальним інтересам ваших клієнтів.

1.2. Клієнт-серверна модель додатку

Клієнт-серверна модель додатку – це архітектурна модель, в якій додаток розбитий на дві частини: клієнтську та серверну.

Клієнтська частина додатку – це програмне забезпечення, яке встановлюється на кінцевий пристрій користувача (наприклад, комп'ютер, смартфон або планшет). Клієнтська частина зазвичай забезпечує інтерфейс для користувача, який дозволяє йому взаємодіяти з серверною частиною.

Серверна частина додатку – це програмне забезпечення, яке встановлюється на сервері. Серверні застосунки відповідають за збереження даних, обробку запитів користувачів та надання відповідей клієнтській частині додатку. [1]

У клієнт-серверному додатку клієнти взаємодіють з сервером за допомогою мережевого з'єднання, такого як Інтернет або локальна мережа. Клієнти можуть відправляти запити на сервер і отримувати від нього відповіді, які можуть бути даними, результатами обробки або повідомленнями про помилки.

Клієнт-серверна модель додатку є дуже поширеною і використовується в багатьох різних сферах, включаючи технології, комунікації, фінанси та інші.

Ця модель має декілька переваг. По-перше, вона дозволяє розподіляти завдання між клієнтом та сервером, що дозволяє досягти більшої ефективності та швидкодії. По-друге, вона забезпечує більшу безпеку, оскільки клієнти не мають прямого доступу до даних на сервері та можуть взаємодіяти з ним лише через визначений протокол.

						Аркуш
						7
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-04.БР	

Однак, ця модель також має деякі недоліки, такі як можливість затримок в мережі, які можуть призвести до поганого досвіду користувача, а також збільшення складності додатку з точки зору розробки та управління.

У випадку ресторанного додатку, клієнтська частина може мати додаткові функції, такі як відстеження статусу замовлення та сповіщення про зміни у замовленні. Серверна частина може також забезпечувати аналітичні функції, які дозволяють власникам ресторанів аналізувати замовлення та піддавати їх обробці, щоб покращити процес.

У контексті розробки клієнт-серверного додатку для ресторану італійської кухні, клієнтська частина може включати мобільний додаток для замовлення їжі та резервування столиків, а також веб-інтерфейс для здійснення замовлень через браузер. Серверна частина додатку забезпечує зберігання меню ресторану, обробку замовлень, керування резерваціями та надання відповідей клієнтам. Використання клієнт-серверної моделі дозволяє ефективно забезпечити зручну та швидку взаємодію між клієнтами та рестораном, покращити обслуговування гостей і підвищити задоволеність клієнтів.

Клієнт-серверний додаток для ресторану італійської кухні має велике практичне значення. Він дозволяє ресторану оптимізувати процеси замовлення та обслуговування, спростити резервування столиків і підвищити ефективність роботи персоналу. Крім того, такий додаток надає зручний спосіб взаємодії з клієнтами, дозволяючи їм замовляти їжу із зручного для них пристрою та отримувати інформацію про акції та спеціальні пропозиції. Результати розробки клієнт-серверного додатку виявляють потенціал автоматизації та покращення роботи ресторану, сприяючи залученню нових клієнтів і підвищенню рівня задоволеності вже існуючих.

Клієнт-серверна модель застосунок дає змогу розділити застосунок на дві частини - клієнтську та серверну - для забезпечення ефективної взаємодії

						Аркуш
						8
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-04.БР	

між користувачем і сервером. Клієнтська частина встановлюється на кінцевому пристрої користувача і забезпечує зручний інтерфейс для взаємодії з додатком. Серверна частина встановлюється на сервері та відповідає за зберігання даних, обробку запитів і надання відповідей клієнту. Така архітектурна модель дає змогу ефективно розподіляти завдання між клієнтом і сервером, сприяючи прискоренню та підвищенню ефективності роботи додатків. Вона також забезпечує більшу безпеку, оскільки клієнт не має прямого доступу до даних на сервері, а взаємодіє з ним за певним протоколом.

У контексті ресторанних додатків модель клієнт-сервер забезпечує зручну взаємодію між клієнтом і рестораном. На клієнтській стороні застосунку користувачі можуть замовляти їжу, резервувати столики та відстежувати стан своїх замовлень. Серверна сторона відповідає за обробку цих запитів, зберігання меню ресторану і надання відповідей клієнтам. Використовуючи модель клієнт-сервер, ресторани можуть оптимізувати процес замовлення, поліпшити обслуговування гостей і залучити нових клієнтів. Такі додатки також забезпечують зручний спосіб взаємодії з клієнтами і допомагають підвищити задоволеність наявних клієнтів.

У результаті, модель клієнт-сервер застосунку є ефективним і безпечним способом організації взаємодії між клієнтом і сервером. У контексті ресторанних застосунків це допомагає поліпшити обслуговування, оптимізувати процеси та забезпечити зручну взаємодію з користувачами.

1.3. Загальна структура сайту

Сайт є набором веб-сторінок, які містять текст, зображення, відео та інші елементи, які доступні через Інтернет за допомогою браузера. Сайти можуть мати різну тематику та функціональність, такі як інформаційні, рекламні, електронні комерції, блоги та інші.

						Аркуш
						9
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-04.БР	

Основними компонентами сайту є HTML-код (головна структура та контент), CSS-стилі (візуальне оформлення та дизайн) та JavaScript (інтерактивність та додатковий функціонал). Кожен сайт має унікальну адресу URL, яка ідентифікує його в Інтернеті.

Сайти можуть бути статичними (інформація не змінюється без редагування коду) або динамічними (інформація оновлюється автоматично або через взаємодію з користувачем). Динамічні сайти можуть використовувати бази даних та програмування на стороні сервера для генерації сторінок на льоту.

Однією з головних переваг сайтів є їх доступність. Вони можуть бути доступні з будь-якого місця, де є доступ до Інтернету. Крім того, вони є дешевшим та ефективнішим інструментом для реклами та маркетингу, порівняно з традиційними методами.

Сайти є важливим елементом сучасного інтернету та їх функціональність безперечно розширюється з року в рік. На додаток до основних компонентів, згаданих у попередньому тексті, сайти можуть містити інші елементи, такі як мультимедійні презентації, анімації, фотогалереї та інші, що поліпшують відображення контенту та користувацький досвід.

Також варто відзначити, що сайти можуть бути розроблені з різними мовами програмування та фреймворками, що впливає на їх функціональність та продуктивність. Наприклад, сайти на JavaScript можуть бути більш швидкими та масштабованими порівняно зі статичними HTML-сторінками.

Нарешті, слід зазначити, що сайти можуть бути важливими для бізнесу та брендування. Електронна комерція та інтернет-магазини надають можливість продажу товарів та послуг онлайн, а блоги та форуми є важливим інструментом для спілкування та обміну інформацією між людьми зі спільними інтересами.

						Аркуш
						10
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-04.БР	

Веб-сайти є важливим засобом комунікації та обміну інформацією в сучасному суспільстві. Вони дають змогу компаніям, організаціям і приватним особам донести свої ідеї до широкої аудиторії. Сайти також є місцем для розміщення важливої інформації, такої як контактні дані, години роботи, ціни та умови.

Для підприємств сайти відіграють важливу роль у створенні їхньої онлайн-присутності та бренду: електронна комерція та інтернет-магазини дають змогу клієнтам купувати безпосередньо через Інтернет, пропонуючи зручність і широкий асортимент товарів і послуг. Блоги та форуми створюють платформу для людей зі спільними інтересами, де вони можуть ділитися своїми думками, ідеями, досвідом і знаннями, стимулюючи активну спільноту і взаємодію.

Розробка веб-сайтів з використанням різних мов програмування і фреймворків значно розширює можливості для створення функціональних і продуктивних веб-додатків. Наприклад, JavaScript можна використовувати для створення швидких, масштабованих веб-сайтів, що забезпечують ефективну обробку запитів і взаємодію з базами даних.

Оскільки доступ до Інтернету зараз є практично в кожного, веб-сайти стають потужними інструментами для спілкування, реклами, комерції та обміну інформацією. Використовуючи веб-сайт, компанії та організації можуть залучати нових клієнтів, збільшувати продажі та покращувати обслуговування клієнтів. Для користувачів веб-сайти не тільки забезпечують доступ до потрібної їм інформації, а й надають можливості для спілкування та співпраці в онлайн-овому середовищі.

Загалом, сайти відіграють важливу роль у сучасному суспільстві як засіб спілкування, торгівлі, розваги та обміну інформацією. Їхнє значення постійно зростає, і майбутнє розвитку веб-сайтів обіцяє нові можливості та інновації.

						Аркуш
						11
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-04.БР	

1.4. Технічне завдання на програмне забезпечення

Головним бенефіціаром системи є власник ресторану або ресторанної мережі, який отримує можливість автоматизувати процес замовлення їжі та керувати ним зручним способом. Застосування ресторанного додатку може допомогти збільшити продажі ресторану та покращити задоволеність клієнтів.

Потенційними користувачами системи є клієнти ресторану, які можуть скористатися додатком для замовлення їжі з мобільного пристрою або веб-сайту. Також системою можуть користуватися працівники ресторану, які зможуть приймати замовлення та керувати їхнім станом через серверну частину додатку.

Аналіз вимог є ключовим етапом у розробці будь-якого програмного забезпечення, в тому числі й клієнт-серверного додатку ресторану італійської кухні. Оскільки мета дослідження полягає у розробці цього додатку, важливо ретельно проаналізувати вимоги до нього.

Основні вимоги до клієнт-серверного додатку ресторану італійської кухні можна сформулювати наступним чином:

1. Функціональність:

- Клієнт повинен мати можливість здійснювати замовлення страв та напоїв з меню ресторану.
- Клієнт повинен мати можливість здійснювати замовлення столика в ресторані.
- Клієнт повинен мати можливість здійснювати запит на доставку їжі з ресторану.
- Клієнт повинен мати можливість оплачувати замовлення з використанням різних методів оплати.
- Клієнт повинен мати можливість додавати відгуки про ресторан та оцінювати якість обслуговування.

						Аркуш
						12
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-04.БР	

2. Ефективність:

- Додаток повинен працювати швидко та без перебоїв.
- Додаток повинен мати стійкість до великої кількості запитів та відвідувань.

3. Безпека:

- Додаток повинен забезпечувати безпеку та конфіденційність користувацької інформації.
- Додаток повинен мати механізми захисту від зловживань та атак.

4. Зручність:

- Додаток повинен мати зручний та інтуїтивно зрозумілий інтерфейс користувача.
- Додаток повинен бути доступним на різних платформах та пристроях.

Вимоги до функціональності додатку дуже важливі для забезпечення задоволення потреб користувачів. Клієнт повинен мати можливість здійснювати замовлення страв та напоїв з меню ресторану, здійснювати замовлення столика в ресторані, а також здійснювати запит на доставку їжі з ресторану. Оплата замовлення з використанням різних методів оплати також є важливою функцією.

Ефективність додатку теж є важливою вимогою. Додаток повинен працювати швидко та без перебоїв, а також мати стійкість до великої кількості запитів та відвідувань. Це допоможе забезпечити користувачам позитивний досвід використання додатку.

Нарешті, безпека є важливою вимогою для будь-якого додатку. Додаток повинен забезпечувати безпеку та конфіденційність користувацької інформації, а також мати механізми захисту від зловживань та атак.

Узагальнюючи, всі ці вимоги є важливими для створення успішного клієнт-серверного додатку ресторану італійської кухні, що забезпечить

						Аркуш
						13
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-04.БР	

користувачам зручність, доступність, функціональність, ефективність та безпеку.

Вимоги до методичного забезпечення для розробки клієнт-серверного додатку ресторану італійської кухні можуть включати наступні пункти:

Чіткість і зрозумілість інструкцій та рекомендацій: методичне забезпечення повинно бути написане таким чином, щоб розробник міг легко розуміти, які кроки потрібно виконувати для розробки додатку. Інструкції та рекомендації повинні бути логічно організовані та відображати послідовність дій.

Повнота: методичне забезпечення повинно охоплювати всі аспекти розробки додатку, включаючи аналіз вимог, проектування, реалізацію, тестування та випуск виробу. Всі важливі етапи процесу розробки повинні бути детально описані та пояснені.

Структурованість: методичне забезпечення повинно мати логічну структуру та бути організованим у вигляді окремих розділів, які відображають окремі етапи процесу розробки додатку. Кожен розділ повинен мати зрозумілу назву та відповідну структуру.

Актуальність: методичне забезпечення повинно бути актуальним та відповідати останнім тенденціям та рекомендаціям у галузі розробки програмного забезпечення. Воно повинно охоплювати останні версії технологій та програмних інструментів, що використовуються для розробки додатків.

Практичність: методичне забезпечення повинно бути практичним та допомагати розробнику виконувати певні завдання. Інструкції та рекомендації повинні бути зорієнтовані на покращення процесу розробки та ефективного використання клієнт-серверного додатку ресторану італійської кухні.

						Аркуш
						14
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-04.БР	

1.5. Висновки до розділу 1

Під час розроблення клієнт-серверних застосунків для ресторанів важливо враховувати різноманітність італійської кухні, її походження та кулінарну історію. Крім того, застосунок може надавати інформацію про кухню, характерну для певного регіону, щоб клієнти могли більше дізнатися про культурний фон і традиції Італії. Такий застосунок може забезпечити автентичний і задовільний досвід відвідування ресторану, розширити знання клієнтів про італійську культуру та сприяти успіху закладу.

Технологічний розвиток та змінювані потреби споживачів ставлять нові вимоги до ресторанів, такі як швидкість обслуговування, онлайн-замовлення та безготівкові платежі. Розробка клієнт-серверного додатку може задовольнити ці вимоги, забезпечуючи зручний та швидкий доступ до послуг ресторану.

Клієнт-серверна модель додатку для італійського ресторану дає змогу оптимізувати процеси бронювання та обслуговування, поліпшити комунікацію з клієнтами та підвищити задоволеність користувачів. Вона забезпечує ефективну взаємодію між клієнтом і рестораном, даючи їм змогу зручно отримувати необхідну інформацію. Розробка клієнт-серверних додатків для ресторанів - важливий крок у використанні сучасних технологій для поліпшення ресторанного бізнесу та забезпечення задоволеності клієнтів.

Отже, розробка клієнт-серверного додатку для ресторану італійської кухні є актуальною і має потенціал покращити роботу ресторану, задовольнити потреби клієнтів та підвищити його конкурентоспроможність на ринку.

						Аркуш
						15
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-04.БР	

РОЗДІЛ 2

ПРОГРАМНІ ЗАСОБИ ПРОЄКТУВАННЯ КЛІЄНТ-СЕРВЕРНОГО ДОДАТКУ

2.1. Опис архітектури додатку

Архітектура - це принцип організації компонентів у системі. До них належать кількість, якість, інтерфейси та протоколи взаємодії. Від архітектури залежать ціна підтримки та розробки нової функціональності, трудовитрати на побудову всієї системи з використанням заданої архітектури. Іншими словами, від архітектури залежить найважливіший параметр розроблення – параметр вартості розроблення. Побічно він також визначає можливість повторного використання.

Архітектура додатку може варіюватися в залежності від його конкретного функціоналу та вимог. Загальна структура та компоненти, які можуть бути присутніми в архітектурі веб-додатку:

1. Клієнтська частина:

- **Фронтенд (frontend):** Це частина додатку, яка взаємодіє з користувачем. Вона відповідає за відображення інтерфейсу, збирання та обробку користувацьких даних. Фронтенд може бути розроблений за допомогою різних технологій, таких як HTML, CSS, JavaScript, фреймворки, такі як React, Angular або Vue.js.
- **Взаємодія з сервером (server communication):** Цей компонент відповідає за обмін даними між клієнтом і сервером. Він може включати AJAX-запити, веб-сокети або інші протоколи комунікації, щоб передавати та отримувати дані з сервера.

2. Серверна частина:

					<i>ДТЕУ 121 07-04.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		03.03.23	Клієнт-серверний додаток ресторану італійської кухні	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Рзаєва С.Л.		03.03.23		P2	16	36
Гарант		Рзаєва С.Л.		03.03.23		Факультет інформаційних технологій 4 курс, 7 група		
Розробив		Вишненко В.В.		03.03.23				
					<i>Програмні засоби проєктування клієнт-серверного додатку</i>			

- Бекенд (backend): Це центральна частина додатку, яка обробляє логіку бізнес-процесів, зберігає та управляє дані. Бекенд може бути розроблений за допомогою мов програмування, таких як Python, Java, PHP або Node.js. Він включає в себе роутинг, контролери, моделі, базу даних, систему аутентифікації та авторизації, взаємодію зі сторонніми службами тощо.
- База даних (database): Вона використовується для зберігання та управління даними, необхідними для роботи додатку. Популярні бази даних, такі як MySQL, PostgreSQL або MongoDB, можуть бути використані залежно від вимог та характеру додатку. Вони забезпечують зберігання даних у структурованому форматі та надають можливості для оптимізації та розширення функціоналу.

3. Інфраструктура:

- Хостинг (hosting): Додаток може бути розгорнутий на фізичних або віртуальних серверах, хмарних платформах або спеціалізованих хостинг-провайдерах. Вибір хостингу залежить від масштабу проекту, вимог до продуктивності та доступності.
- Система контролю версій (version control system): Використання системи контролю версій, такої як Git, дозволяє керувати та відстежувати зміни в коді додатку, спільно працювати над проектом та легко відновлювати попередні версії.
- Серверні сервіси: Для розгортання та управління додатком можуть використовуватись різні серверні сервіси, такі як веб-сервери (наприклад, Apache або Nginx), контейнеризація (наприклад, Docker) або оркестрація контейнерів (наприклад, Kubernetes).

						Аркуш
						17
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-04.БР	

Це загальна структура архітектури веб-додатку, але конкретна реалізація може відрізнятись в залежності від потреб і вимог проекту. Важливо враховувати принципи модульності, масштабованості та безпеки при проектуванні архітектури додатку.

2.2. Моделі та діаграми: для чого і коли створюються

Моделі та діаграми є важливими інструментами для розробки програмного забезпечення, які допомагають відображати складні структури та процеси, що відбуваються в системі. Вони використовуються для уточнення вимог до системи, проектування архітектури, розробки та тестування програмного забезпечення.

Моделі та діаграми створюються на ранніх етапах розробки програмного забезпечення, коли необхідно уточнити вимоги до системи та проектувати її архітектуру. Вони допомагають уникнути помилок та недорозумінь, що можуть виникнути на етапі розробки програмного забезпечення.

Моделі та діаграми також полегшують комунікацію між учасниками проекту. Вони служать засобом візуалізації та спільного розуміння складних концепцій, що допомагає уникнути недорозумінь та вирішити потенційні конфлікти в ранніх етапах розробки.

Уніфікована мова моделювання (Unified Modeling Language - UML) – це стандартний інструмент для розроблення «креслень» ПЗ (програмного забезпечення), що використовується для візуалізації, специфікації, конструювання й документування артефактів ПЗ.

UML - це стандартний інструмент для створення креслень програмного забезпечення. Одна модель системи ніколи не є абсолютно достатньою. Для розуміння більшості систем потрібна ціла взаємозалежна модель. У випадку програмного забезпечення мова моделювання повинна мати засоби для

						ДТЕУ 121 07-04.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			18

опису архітектури системи з різних точок зору протягом усього життєвого циклу розроблення: словник і правила UML описують, як створювати та читати чітко визначені моделі, але не кажуть, коли та які саме моделі слід створювати. У ньому нічого не говориться про те, коли і які саме моделі мають бути створені. Це проблема для всього процесу розроблення програмного забезпечення. Добре організований процес має підказати, які результати роботи необхідні, які ресурси потрібні для їхнього створення та управління ними, як оцінювати роботу з ними і як управляти проектом загалом.

UML - це мова візуалізації. Деякі речі найкраще висловлювати безпосередньо в коді мови програмування. Текст програми - це найпряміший і найкоротший спосіб запису виразів та алгоритмів. Реалізація проекту - це майже те саме, що й написання коду. Але все ж програміст фактично бере участь у моделюванні, хоча й неформально.

Однак при цьому виникає кілька проблем із розробкою:

- По-перше, під час обговорення таких концептуальних моделей з іншими учасниками розроблення можуть виникнути помилки та непорозуміння, якщо тільки всі учасники обговорення не говорять однією мовою.
- По-друге, під час розроблення програмного забезпечення багато речей важко змоделювати мовою програмування, що тільки-но написана (ієрархії класів можна зрозуміти, вивчивши код кожного класу в ієрархії. Однак вивчення коду системи не дає повної картини, включно з фізичним розподілом об'єктів і можливими переходами).
- По-третє, якщо розробник, який написав цей код, так і не реалізував модель, що існує в його голові, інформація про модель може бути втрачена назавжди.

						ДТЕУ 121 07-04.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			19

2.3. Еталонна модель сайту ресторану

Еталонна модель сайту ресторану є базовою моделлю, яка дозволяє визначити загальну структуру та функціональність сайту. Вона включає в себе основні елементи, такі як головне меню, каталог страв, можливість замовлення та доставки, резервування столиків, контактну інформацію та інші ключові функції, необхідні для забезпечення зручного та ефективного користувацького досвіду.

Використання еталонної моделі веб-сайту спрощує процес розробки. Це пов'язано з тим, що еталонна модель сайту - це готовий шаблон, який містить основну структуру і компоненти, які можна використовувати для розробки нового сайту. Це дає змогу заощадити багато часу та зусиль під час створення нового сайту.

Еталонна модель сайту може використовуватися як візуальний засіб для зрозуміння та комунікації між розробниками, дизайнерами та іншими учасниками проекту. Вони можуть допомогти досягти спільного розуміння структури та функціональності сайту, покращити співпрацю та уникнути непорозумінь.

Основні складові еталонної моделі сайту ресторану можуть включати:

- Головна сторінка: На головній сторінці розміщується заголовок, фотографії або слайдер зображень, вітання від ресторану та можливість швидкого доступу до ключових функцій, таких як меню, замовлення та резервування.
- Головне меню: У цьому розділі розміщується перелік основних категорій страв або напоїв, що пропонуються в ресторані. Кожна категорія може мати підкатегорії або додаткові деталі, такі як опис страв, ціни, фотографії тощо.
- Каталог страв: Це розділ, де користувач може переглядати повний список страв, їх опис, ціни та інші характеристики. Користувачі можуть

						ДТЕУ 121 07-04.БР	Аркуш
							20
Зм.	Аркуш	№ докум	Підпис	Дата			

використовувати фільтри або пошукову функцію, щоб знайти конкретну страву або категорію.

- **Замовлення та доставка:** Цей розділ дозволяє користувачам обрати страву, додати її до кошика, вказати адресу доставки та провести оплату. Крім того, можуть бути доступні інші опції, такі як вибір часу доставки або вказання особливих вимог до замовлення.
- **Резервування столиків:** Якщо ресторан має можливість резервування столиків, то на сайті може бути розділ, де користувачі можуть обрати дату, час та кількість гостей для резервування. Також можуть бути вказані додаткові вимоги або примітки.
- **Контактна інформація:** На сайті ресторану повинна бути доступна контактна інформація, така як номер телефону ресторану, електронна пошта, адреса розташування та години роботи. Крім того, можна надати форму зворотного зв'язку, яка дозволяє користувачам надсилати запитання, коментарі або пропозиції безпосередньо через веб-сайт.

Ці основні складові еталонної моделі сайту ресторану можуть бути доповнені або змінені відповідно до конкретних потреб та особливостей ресторану. Додаткові функції, такі як онлайн-бронювання столиків, програма лояльності, огляди та рейтинги страв, можуть також бути включені в модель відповідно до стратегії ресторану та вимог клієнтів.

2.3. Діаграма стану бронювання столика в ресторані

Діаграма станів - це особливий вид графа, який являє собою автомат. Вершини графа - це можливі стани автомата, представлені відповідними графічними символами, а дуги позначають переходи зі стану в стан. Діаграми станів можуть бути вкладені одна в одну для більш детального представлення окремих елементів моделі.

						ДТЕУ 121 07-04.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			21

Діаграми станів використовуються для опису поведінки складних об'єктів. Діаграми станів показують, як об'єкти переходять з одного стану в інший. Очевидно, що діаграми станів використовуються для моделювання динамічних аспектів системи.

Діаграми станів корисні під час моделювання життєвого циклу об'єкта. Діаграми станів відрізняються від інших діаграм тим, що вони описують процес зміни стану одного екземпляра класу, тобто одного об'єкта, і описують реактивні об'єкти, тобто об'єкти, поведінка яких характеризується їхньою реакцією на зовнішні події. Концепція життєвого циклу застосовується до реактивних об'єктів, поточний стан яких визначається їхнім минулим станом. Однак діаграми станів важливі не тільки для опису динаміки окремого об'єкта. Діаграми станів можна використовувати для проєктування виконуваних систем за допомогою прямого і зворотного інжинірингу.

Діаграми станів найчастіше використовують для опису поведінки окремих об'єктів, але їх також можна застосовувати для визначення функціональності інших компонентів моделі, таких як сценарії використання, актори, підсистеми, операції та методи.

Діаграма стану бронювання столика є інструментом, який відображає різні стани, які можуть виникнути під час процесу бронювання столика в ресторані. Ця діаграма допомагає візуалізувати послідовність дій та зміни станів, пов'язані з бронюванням столика.

Діаграми стану корисні для моделювання різноманітних процесів і систем, таких як керування станом програмного забезпечення, бізнес-процеси, системи автоматизації та мережеві протоколи. Вони допомагають зрозуміти логіку вашої системи, виявити можливі проблеми та підвищити ефективність.

						ДТЕУ 121 07-04.БР	Аркуш
							22
Зм.	Аркуш	№ докум	Підпис	Дата			

Діаграми станів необхідні, оскільки їх використовують аналітики, проєктувальники та розробники для вивчення поведінки об'єктів у системі. Діаграми класів і відповідні діаграми об'єктів показують тільки статичний стан системи. Вони представляють ієрархії та відносини, з яких ми можемо дізнатися про список можливих варіантів поведінки системи, але нічого про деталі динамічної поведінки(рис.2.1.).

Однак розробникам необхідно розуміти поведінку об'єктів. Це пов'язано з тим, що їхня робота полягає в реалізації цієї поведінки в програмному забезпеченні. Недостатньо спроектувати об'єкт; фахівці мають переконатися, що об'єкт виконує свою функцію. Діаграми станів надають повну інформацію про бажану поведінку. Чітке розуміння поведінки об'єкта підвищує ймовірність того, що команда розробників створить систему, яка відповідає вимогам.



Рисунок 2.1. Діаграма стану бронювання столика

Джерело: побудовано автором

У цій діаграмі ми маємо п'ять станів: "Перевірка", "Очікування", "Відміна бронювання", "Виконання бронювання" та "Бронювання". Стрілки показують переходи між станами. Наприклад, зі стану "Перевірка" можна

						Аркуш
						23
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-04.БР	

перейти до "Виконання бронювання", а з " Виконання бронювання" - до "Бронювання" або "Відміна бронювання". Залежно від умов та дій, можуть виникати інші переходи між станами.

2.4. Діаграма класів сайту ресторану

Діаграма класів показує структуру системи та зв'язки між класами. Вони використовуються для визначення базових класів, атрибутів, методів і успадкування між класами. Це базова схема для побудови коду програми. Внутрішня структура системи створюється за допомогою діаграм класів, що описують успадкування класів і відносні позиції. Тут ми опишемо логічне представлення системи. Це логічно, оскільки класи — це лише простори, у яких пізніше визначаються фізичні об'єкти. Діаграми класів - це терміни класів об'єктно-орієнтованого програмування, що використовуються для представлення статичної структури моделі системи. Діаграми класів можуть зображати, зокрема, різні відносини між окремими сутностями в галузі, що цікавить, такими як об'єкти та підсистеми, і описувати їхню внутрішню структуру та типи відносин. Діаграма не надає інформацію про часові аспекти системи. У цьому контексті подальшим розвитком концептуальної моделі проекрованої системи є діаграма класів.

Діаграма класів являє собою граф, вершинами якого є елементи типу "класифікатор", пов'язані між собою різними типами структурних відносин. Варто зазначити, що діаграми класів можуть також містити інтерфейси, пакети, відносини і навіть окремі екземпляри об'єктів і відносин. Говорячи про таку діаграму, ми маємо на увазі статичну структурну модель проекрованої системи. Тому діаграму класів можна розглядати як незалежне від часу графічне представлення структурних зв'язків логічної моделі такої системи.

						ДТЕУ 121 07-04.БР	Аркуш
							24
Зм.	Аркуш	№ докум	Підпис	Дата			

Діаграма класів складається з набору елементів, які в сукупності представляють декларативні знання про предметну область. Ці знання інтерпретуються в термінах основних понять мови UML, таких як класи, інтерфейси та їхні зв'язки з компонентами. Водночас, окремі компоненти цієї діаграми можуть утворювати пакет для представлення більш загальної моделі системи. Якщо діаграма класів є частиною пакета, то її компоненти мають відповідати елементам цього пакета(рис.2.2.).

У загальному випадку пакет статичних структурних моделей може бути представлений у вигляді однієї або декількох діаграм класів. Декомпозиція подання на окремі діаграми робиться для зручності та для графічної візуалізації структурних зв'язків між цільовими областями. Компоненти діаграми відповідають елементам статичної семантичної моделі. Модель системи має відповідати внутрішній структурі класів, описаних мовою UML.

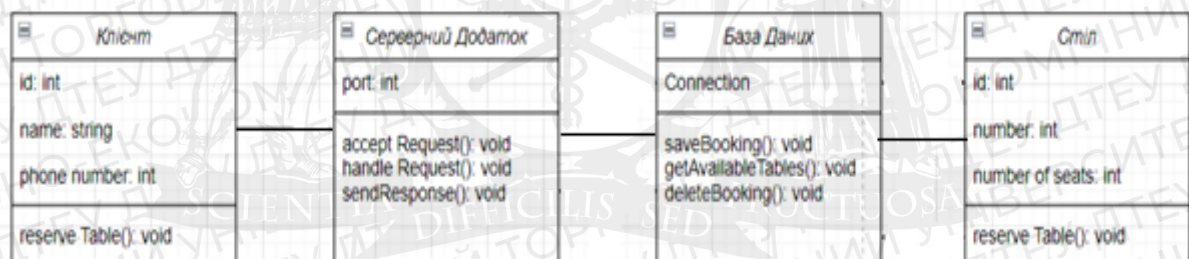


Рисунок 2.2. Діаграма класів
Джерело: побудовано автором

2.5. Висновки до розділу 2

Отже, підсумовуючи вище викладене, можна виділити важливі аспекти. Вибір правильного програмного забезпечення - критичний етап проекту. Необхідно враховувати вимоги проекту, функціональність,

						Аркуш
						25
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-04.БР	

масштабованість, вартість, підтримку та інші фактори. Важливо ретельно проаналізувати потреби проєкту і вивчити різні програмні інструменти, щоб знайти найкращий варіант. Під час розроблення клієнт-серверних додатків також необхідно враховувати взаємодію між клієнтською та серверною частинами. Під час проєктування архітектури необхідно враховувати поділ функцій між клієнтом і сервером, вибір протоколів зв'язку, а також питання безпеки та масштабованості. Також необхідно визначити, як обробляються помилки і як зберігаються дані.

З огляду на ці міркування, розробка серверної частини з використанням мов програмування HTML, CSS і JavaScript вимагає глибокого розуміння веб-технологій і взаємодії між клієнтом і сервером для створення функціонального, безпечного та ефективного сайту ресторану з резервуванням столиків.

							Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-04.БР		26

РОЗДІЛ 3

РЕАЛІЗАЦІЯ КЛІЄНТ-СЕРВЕРНОГО ДОДАТКУ

3.1. Вибір програмного забезпечення для побудови та налагодження бази даних та її структури

База даних є важливим компонентом будь-якого веб-сайту, оскільки вона зберігає дані, необхідні для правильної роботи сайту. Для ресторану база даних може містити інформацію про меню, замовлення та резервації столиків.

Структура бази даних може бути організована в табличному вигляді з використанням різних ключів та зв'язків між таблицями. Наприклад, таблиця "страви" може мати відношення з таблицею "замовлення", що дозволяє легко відстежувати, які страви були замовлені для кожного замовлення.

Для побудови та налагодження бази даних можна використовувати різне програмне забезпечення, таке як MySQL, PostgreSQL або MongoDB. Вибір програмного забезпечення залежить від потреб проекту, а також від технічних вимог до бази даних.

У цьому розділі розглянемо процес створення бази даних клієнт-серверної програми для італійського ресторану. База даних є важливою частиною системи, оскільки вона забезпечує ефективне керування діяльністю ресторану.

Необхідно визначити інформацію, яка буде зберігатися в базі даних. В даному випадку це дані про столики, бронювання та користувачів. Дані про столики міститимуть таку інформацію як ідентифікатори столів, кількість місць, розташування в ресторані та можливість бронювання. Дані про бронювання в свою чергу включатимуть інформацію про дату та час

					<i>ДТЕУ 121 07-04.БР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		14.04.23	Клієнт-серверний додаток ресторану італійської кухні	Стадія	Аркуш	Аркушів
Керівник		Рзаєва С.Л.		14.04.23		РЗ	27	36
Гарант		Рзаєва С.Л.		14.04.23		Факультет інформаційних технологій 4 курс, 7 група		
Розробив		Вищенко В.В.		14.04.23				
					<i>Реалізація клієнт-серверного додатку</i>			

бронювання, кількість осіб, ім'я клієнта та контактні дані. Дані про користувачів міститимуть таку інформацію як відомості про зареєстрованих користувачів, а саме ім'я, електронна адреса, пароль, історія замовлень(таб.3.1.,3.2.,3.3.)

Таблиця 3.1.

«Столики»

Столики		CREATE TABLE "Столики" ("Id" INTEGER NOT NULL UNIQUE, "Кількість місць" INTEGER NOT NULL, "Розташування" TEXT NOT NULL, "Можливість бронювання" INTEGER NOT NULL, PRIMARY KEY("Id"))
Id	INTEGER	"Id" INTEGER NOT NULL UNIQUE
Кількість місць	INTEGER	"Кількість місць" INTEGER NOT NULL
Розташування	TEXT	"Розташування" TEXT NOT NULL
Можливість бронювання	INTEGER	"Можливість бронювання" INTEGER NOT NULL

Джерело: побудовано автором

Таблиця 3.2.

«Бронювання»

Бронювання		CREATE TABLE "Бронювання" ("ID" INTEGER NOT NULL UNIQUE, "ID столика" INTEGER NOT NULL UNIQUE, "Дата та час" TEXT, "Кількість осіб" INTEGER, "Ім'я клієнта" TEXT, "Контактні дані" TEXT, PRIMARY KEY("ID"))
ID	INTEGER	"ID" INTEGER NOT NULL UNIQUE
ID столика	INTEGER	"ID столика" INTEGER NOT NULL UNIQUE
Дата та час	TEXT	"Дата та час" TEXT
Кількість осіб	INTEGER	"Кількість осіб" INTEGER
Ім'я клієнта	TEXT	"Ім'я клієнта" TEXT
Контактні дані	TEXT	"Контактні дані" TEXT

Джерело: побудовано автором

Таблиця 3.3.

«Користувачі»

Користувачі		CREATE TABLE "Користувачі" ("ID" INTEGER NOT NULL UNIQUE, "Ім'я" INTEGER NOT NULL, "Електронна адреса" TEXT NOT NULL UNIQUE, "Пароль" INTEGER NOT NULL, "Історія замовлень" INTEGER, PRIMARY KEY("ID" AUTOINCREMENT))
ID	INTEGER	"ID" INTEGER NOT NULL UNIQUE
Ім'я	INTEGER	"Ім'я" INTEGER NOT NULL
Електронна адреса	TEXT	"Електронна адреса" TEXT NOT NULL UNIQUE
Пароль	INTEGER	"Пароль" INTEGER NOT NULL
Історія замовлень	INTEGER	"Історія замовлень" INTEGER

Джерело: побудовано автором

Крім того, потрібно враховувати операції, які виконуватимуться з даними, наприклад додавання нових записів, оновлення існуючих записів, видалення та пошук. Наприклад, можливість додавати нові столики до бази даних, оновлювати статус бронювання, видаляти старі записи про бронювання та шукати доступні столики в певний час і кількість людей.

Оскільки сайт ресторану може мати обмежену кількість відвідувачів та не потребує великої складності, а також наявність безкоштовного хостингу,

було вирішено використати базу даних MySQL. Вона є найбільш популярною, має велику кількість документації та зручний інтерфейс управління.

Фізична модель бази даних необхідна для визначення структури даних, оптимізації продуктивності та забезпечення цілісності даних у базі даних. Він визначає таблиці, стовпці, типи даних, індекси та інші фізичні атрибути, які дозволяють ефективно зберігати та обробляти дані. Фізична модель також враховує такі аспекти, як розміщення даних на диску, оптимізація запитів і безпека даних. Часто фізична модель допомагає перевести концептуальну модель бази даних у реалізовану структуру, яку можна використовувати для зберігання та обробки даних у сценаріях реального світу(рис.3.1.).

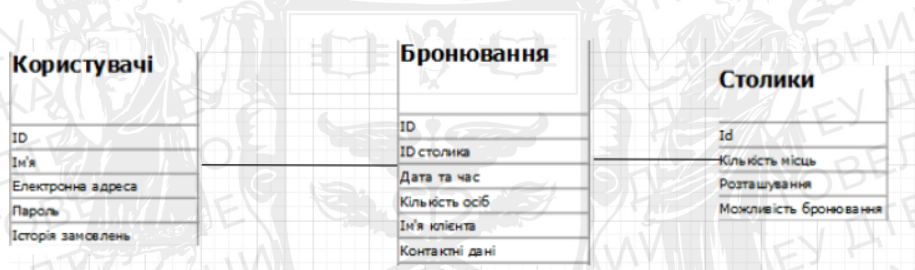


Рисунок 3.1. Логічна модель бази даних

Джерело: побудовано автором

Фізична модель бази даних також охоплює визначення фізичних параметрів бази даних, як-от розмір блоків даних, методи зберігання та індексування, стратегії резервного копіювання та відновлення даних, а також розміщення бази даних на фізичних пристроях зберігання, як-от сервери або хмарні сервіси.

Фізична модель бази даних дає змогу реалізувати логічну модель бази даних у конкретному середовищі. Вона дає змогу оптимізувати продуктивність і ефективність бази даних, забезпечити надійність даних і захистити їх від втрати або пошкодження. Крім того, фізична модель

враховує особливості та обмеження обраного програмного забезпечення бази даних.

Загальне розуміння фізичної моделі бази даних дає змогу розробникам і адміністраторам ефективно керувати та підтримувати базу даних, виконувати оптимізацію запитів і забезпечувати належну роботу системи зберігання даних.

3.2. Використання CSS, HTML та JavaScript

Веб-розробка базується на знанні HTML, CSS та JavaScript. HTML відповідає за структуру сторінки, CSS - за її вигляд, а JavaScript - за динаміку та інтерактивність.

У розділі використано CSS для створення зручного та привабливого інтерфейсу сайту ресторану. Завдяки використанню CSS, можна легко змінювати зовнішній вигляд сайту, додавати нові елементи та взаємодіяти з користувачем(рис.3.2.).

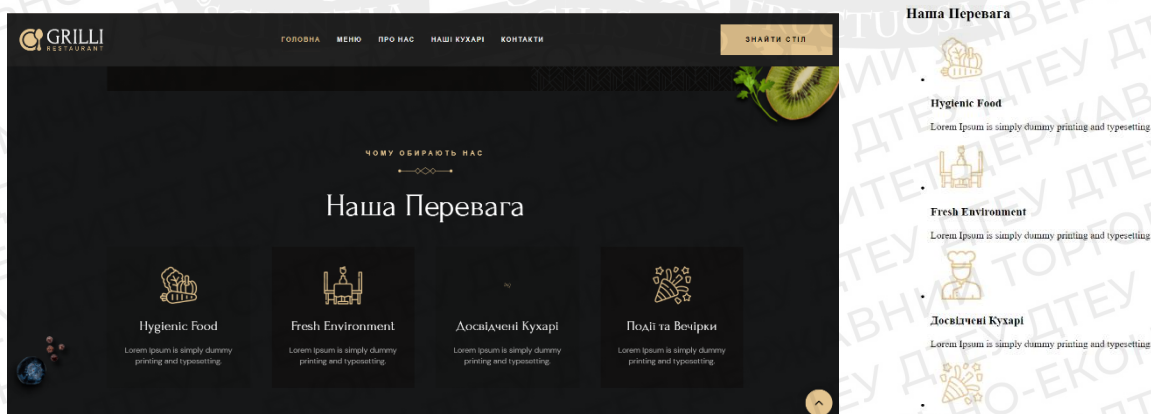


Рисунок 3.2. Сайт з застосуванням CSS та без
Джерело: побудовано автором

						ДТЕУ 121 07-19.БР	Аркуш
							30
Зм.	Аркуш	№ докум	Підпис	Дата			

Було створено стилі, які дозволяють створювати різноманітні елементи інтерфейсу, такі як кнопки, форми, меню, заголовки, фонові зображення та інше. Було використано гнучкий дизайн, щоб сторінка розгорталася оптимально на різних розмірах екранів. Також було додано анімаційні ефекти для підвищення інтерактивності та естетичності сайту.

JavaScript є мовою програмування, яка дозволяє створювати динамічні та інтерактивні елементи на веб-сторінках. У розділі було використано JavaScript для створення галереї зображень страв, що дозволяє користувачеві зручно переглядати фотографії страв ресторану.

Також були створені інтерактивні елементи, такі як випадючі списки, кнопки та форми, що підвищують зручність користування сайтом. Наприклад, на сторінці замовлення користувач може обрати бажану кількість страв, додаткові інгредієнти та інші параметри, які було реалізовано за допомогою випадючих списків та форм(рис.3.3.).

```
1 const parallaxItems = document.querySelectorAll(  
2   "[data-parallax-item]");  
3  
4 let x, y;  
5  
6 window.addEventListener("mousemove",  
7   function(event) {  
8  
9     x = (event.clientX / window.innerWidth *  
10    10) - 5;  
11    y = (event.clientY / window.innerHeight *  
12    10) - 5;  
13  
14    // reverse the number eg. 20 -> -20, -5 -> 5  
15    x = x - (x * 2);  
16    y = y - (y * 2);  
17  
18    for (let i = 0, len =  
19    parallaxItems.length; i < len; i++) {  
20      x = x * Number(parallaxItems[i]  
21      .dataset.parallaxSpeed);  
22      y = y * Number(parallaxItems[i]  
23      .dataset.parallaxSpeed);  
24      parallaxItems[i].style.transform =  
25      `translate3d(${x}px, ${y}px, 0px)`;  
26    }  
27  }  
28  );
```

Рисунок 3.3. Приклад застосування JavaScript

Джерело: побудовано автором

Для забезпечення зручності перегляду страв ресторану була розроблена галерея зображень, яка відображає зображення страв з меню. Ця

						Аркуш
						31
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 07-19.БР	

галерея була реалізована за допомогою JavaScript, що дозволяє користувачеві зручно переглядати картинки, збільшувати їх та перемикатися між зображеннями.

Отже, розробка сайту ресторану з використанням CSS, HTML, JavaScript та бази даних дозволила створити зручний та привабливий інтерфейс для користувачів, що підвищує рівень комфорту та задоволення від відвідування ресторану. Реалізація функціональності сайту дозволяє здійснювати швидке та зручне замовлення страв, а також оновлювати та розширювати меню ресторану з мінімальними зусиллями.

3.3. Розробка форми для бронювання столиків

Бронювання столиків дозволяє ресторанам ефективно планувати свої процеси та оптимально використовувати наявні ресурси, такі як столики та персонал. Завдяки бронюванню столиків ресторани можуть передбачити очікувану кількість гостей, забезпечити правильний розподіл столиків і запропонувати зручність своїм клієнтам.

Бронювання дає змогу спланувати відвідування заздалегідь, особливо в популярних ресторанах або ресторанах з обмеженою кількістю місць. Це дозволяє уникнути ризику нестачі місць у клієнтів і робить їх перебування в ресторані більш комфортним і приємним.

Для забезпечення можливості бронювання столиків було розроблено відповідну форму на сайті ресторану. Користувач має можливість вибрати бажану дату та час, кількість осіб та залишити контактні дані для зв'язку. Інформація з форми автоматично заноситься до бази даних ресторану та надсилається на електронну пошту адміністратора для подальшої обробки і підтвердження бронювання.

						ДТЕУ 121 07-19.БР	Аркуш
							32
Зм.	Аркуш	№ докум	Підпис	Дата			

Також, на сайті було реалізовано можливість обрати час та дату, що дозволяє користувачу зручно вибрати найбільш підходящий варіант. Ця функція доповнює зручне та доступне для клієнтів ресторанів онлайн-середовище для бронювання столиків і перегляду меню. Такий підхід дає змогу ресторанам залучити більше клієнтів, полегшити процес планування візиту та покращити загальне враження від відвідування (рис. 3.4.).

The image shows a dark-themed web form for online reservations. On the left, under the heading 'Бронювання онлайн', there is a sub-heading 'Забронювати +380 994 629 123 або заповніть форму замовлення'. The form contains several input fields: 'Ваше Ім'я', 'Номер Телефону', a dropdown for '1 Персона', a date picker 'ДД,ММ,ГТТГ', and a time picker '08:00'. Below these is a 'Коментар' field and a prominent orange button labeled 'ЗАБРОНЮВАТИ СТОЛИК'. On the right, the section 'Зв'яжіться з нами' displays the phone number '+380 994 629 123', the address 'вулиця Кіото, 19, Київ', and two menu options: 'Ланч' (11:00 - 14:30) and 'Вечеря' (17:00 - 22:00).

Рисунок 3.4. Форма для бронювання столика

Джерело: побудовано автором

3.4. Висновок до розділу 3

У розділі описано важливі компоненти розробки веб-сайту для ресторану. Було вирішено використати базу даних MySQL для зберігання інформації про меню, замовлення та резервації столиків. MySQL обрано через його популярність, доступність документації та зручний інтерфейс управління.

						ДТЕУ 121 07-19.БР	Аркуш
							33
Зм.	Аркуш	№ докум	Підпис	Дата			

CSS, HTML та JavaScript були використані для створення зручного та привабливого інтерфейсу сайту. Завдяки CSS можна змінювати зовнішній вигляд сайту та додавати елементи взаємодії з користувачем. JavaScript додав динаміку та інтерактивність, зокрема галерею зображень страв та інтерактивні елементи на сторінках.

Було розроблено форму для бронювання столиків, де користувач може вказати дату, час, кількість осіб та контактні дані. Інформація з форми зберігається в базі даних та надсилається на електронну пошту адміністратора для обробки.

Загальною метою розробки сайту ресторану було створення зручного та привабливого інтерфейсу для користувачів, спрощення процесу замовлення та бронювання столиків, а також можливості оновлення та розширення меню ресторану. Використання CSS, HTML, JavaScript та бази даних MySQL дозволило досягти вказаних цілей та забезпечити зручність та ефективність використання сайту для відвідувачів.

						ДТЕУ 121 07-19.БР	Аркуш
							34
Зм.	Аркуш	№ докум	Підпис	Дата			

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Вході проведення дослідження теми випускного кваліфікаційного проекту можна зробити наступні висновки:

1. Проведено аналіз сучасного ринку ресторанного бізнесу і виявлено, що ресторани стали не лише місцем для прийняття їжі, але й популярними місцями для проведення зустрічей та ділових переговорів. Зростає попит на високоякісні продукти та послуги в ресторанах.

2. В рамках проекту була успішно створена база даних італійського ресторану. Для цього було визначено інформацію, яку необхідно зберігати в базі даних. Відповідно до цих вимог була побудована фізична модель бази даних, яка складається з таблиць «Таблиці», «Бронювання» та «Користувачі» з відповідними атрибутами для кожної таблиці.

3. Проведений аналіз розкрив наявність значного попиту на бронювання столиків у ресторанах, особливо в популярних ресторанах з високою відвідуваністю. Виявлено, що клієнти цінують зручність та швидкість процесу бронювання столика, а також можливість вибору конкретного часу та кількості осіб.

4. В процесі виконання проекту розроблено клієнт-серверний додаток ресторану італійської кухні.

5. Розроблена система повністю задовольняє потреби клієнтів, надаючи їм зручний та ефективний спосіб бронювання столиків, доступ до актуальної інформації про ресторан, швидке підтвердження бронювання та високу надійність системи.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 07-04.БР</i>			
Зав. каф.	Криворучко О.В.			28.04.23				
Керівник	Рзаєва С.Л.			28.04.23	Клієнт-серверний додаток ресторану італійської кухні	Стадія	Аркуш	Аркушів
Гарант	Рзаєва С.Л.			28.04.23		ВП	35	36
Розробив	Вишненко В.В.			28.04.23		Факультет інформаційних технологій		
						4 курс, 7 група		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

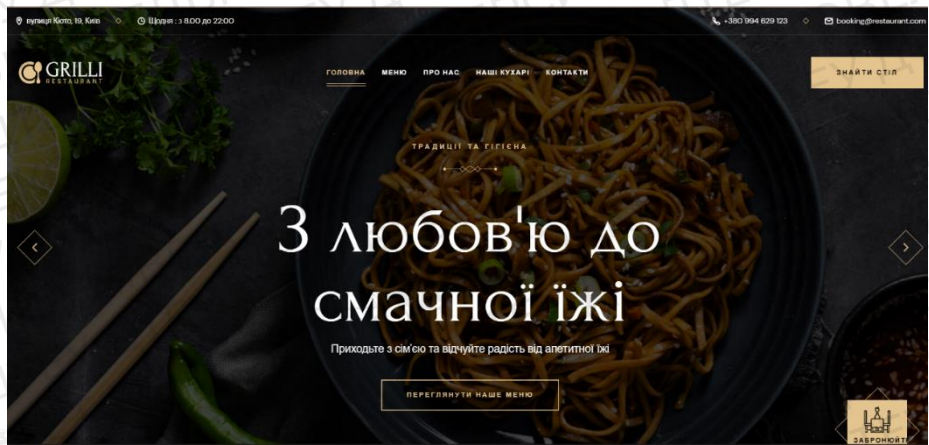
1. Клієнт-сервер [Електронний ресурс]. Режим доступу : <https://developer.mozilla.org/> (дата звернення: 23.12.2022).
2. Desktop vs Browser not just another Thick vs Thin Client debate. [Електронний ресурс] Режим доступу : <https://www.msafocus.com/news/desktop-vs-browser> (дата звернення: 23.12.2022).
3. Study: Italian Cuisine Most Popular Worldwide, Edging Japanese - The Food Institute [Електронний ресурс]. - Режим доступу: <https://foodinstitute.com/focus/study-italian-cuisine-most-popular-worldwide-edging-japanese/> (дата звернення: 23.12.2022).
4. HTML і CSS довідник українською [Електронний ресурс]. - Режим доступу: <https://html-css.co.ua/> (дата звернення: 23.12.2022).
5. Сучасний підручник з JavaScript [Електронний ресурс]. - Режим доступу: <https://uk.javascript.info/> (дата звернення: 23.12.2022).
6. Data Protection For Mobile Client-Server Architectures | by vixentael | Stanfy Engineering Practices | Medium [Електронний ресурс]. - Режим доступу: <https://medium.com/stanfy-engineering-practices/data-protection-for-mobile-client-server-architectures-6e6dcabd871a> (дата звернення: 23.12.2022).

<i>ДТЕУ 121 07-04.БР</i>								
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Клієнт-серверний додаток ресторану італійської кухні <i>Список використаних джерел</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		23.12.22		<i>СВД</i>	36	36
Керівник		Рзаєва С.Л.		23.12.22		Факультет інформаційних технологій 4 курс, 7 група		
Гарант		Рзаєва С.Л.		23.12.22				
Розробив		Вишненко В.В.		23.12.22				

ДОДАТКИ

ДОДАТОК А

Вигляд частини сайту



ДОДАТОК Б

Html код форми для бронювання

```
<section class="reservation">
  <div class="container">
    <div class="form reservation-form bg-black-10">
      <form action="" class="form-left">
        <h2 class="headline-1 text-center">Бронювання онлайн</h2>
        <p class="form-text text-center">
          Забронювати <a href="tel:+88123123456"
            class="link">+380 994 629 123</a> або заповніть форму замовлення
        </p>
        <div class="input-wrapper">
          <input type="text" name="name" placeholder="Ваше Ім'я"
            autocomplete="off" class="input-field">
          <input type="tel" name="phone" placeholder="Номер Телефону"
            autocomplete="off" class="input-field">
        </div>
        <div class="input-wrapper">
          <div class="icon-wrapper">
            <ion-icon name="person-outline" aria-hidden="true"></ion-icon>
            <select name="person" class="input-field">
              <option value="1-person">1 Персона</option>
              <option value="2-person">2 Персони</option>
              <option value="3-person">3 Персони</option>
              <option value="4-person">4 Персони</option>
              <option value="5-person">5 Персон</option>
              <option value="6-person">6 Персон</option>
              <option value="7-person">7 Персон</option>
            </select>
            <ion-icon name="chevron-down" aria-hidden="true"></ion-icon>
          </div>
          <div class="icon-wrapper">
            <ion-icon name="calendar-clear-outline" aria-hidden="true"></ion-icon>
            <input type="date" name="reservation-date"
              class="input-field">
            <ion-icon name="chevron-down" aria-hidden="true"></ion-icon>
          </div>
          <div class="icon-wrapper">
            <ion-icon name="time-outline" aria-hidden="true"></ion-icon>
            <select name="person" class="input-field">
              <option value="08:00am">08 : 00</option>
              <option value="09:00am">09 : 00</option>
              <option value="10:00am">10 : 00</option>
              <option value="11:00am">11 : 00</option>
              <option value="12:00am">12 : 00</option>
              <option value="01:00pm">13 : 00</option>
              <option value="02:00pm">14 : 00</option>
              <option value="03:00pm">15 : 00</option>
              <option value="04:00pm">16 : 00</option>
              <option value="05:00pm">17 : 00</option>
              <option value="06:00pm">18 : 00</option>
            </select>
          </div>
        </div>
      </form>
    </div>
  </div>
</section>
```

```

<option value="07:00pm">19 : 00 </option>
<option value="08:00pm">20 : 00</option>
<option value="09:00pm">21 : 00</option>
</select>
<ion-icon name="chevron-down" aria-hidden="true"></ion-icon>
</div>
</div>
<textarea name="message" placeholder="Коментар"
autocomplete="off" class="input-field"></textarea>
<button type="submit" class="btn btn-secondary">
<span class="text text-1">Забронювати столик</span>
<span class="text text-2" aria-hidden="true">Забронювати столик</span>
</button>
</form>
<div class="form-right text-center" style="background-image: url('/assets/images/form-pattern.png')">
<h2 class="headline-1 text-center">Зв'яжіться з нами</h2>
<p class="contact-label">Забронювати</p>
<a href="tel:+88123123456" class="body-1 contact-number hover-underline">+380 994 629 123</a>
<div class="separator"></div>
<p class="contact-label">Місцезнаходжен</p>
<address class="body-4">
вулиця Кіото, 19, Київ
</address>
<p class="contact-label">Ланч</p>
<p class="body-4">
3 Понеділка по Неділю
<br> 11:00 - 14:30
</p>
<p class="contact-label">Вечеря</p>
<p class="body-4">
3 Понеділка по Неділю
<br> 17:00 - 22:00
</p>
</div>
</div>
</section>

```

ДОДАТОК В

JavaScript код сайту

```
'use strict';  
/**  
 * PRELOAD  
 *  
 * loading will be end after document is loaded  
 */  
const preloader = document.querySelector("[data-preload]");  
window.addEventListener("load", function () {  
  preloader.classList.add("loaded");  
  document.body.classList.add("loaded");  
});  
/**  
 * add event listener on multiple elements  
 */  
const addEventOnElements = function (elements, eventType, callback) {  
  for (let i = 0, len = elements.length; i < len; i++) {  
    elements[i].addEventListener(eventType, callback);  
  }  
}  
/**  
 * NAVBAR  
 */  
const navbar = document.querySelector("[data-navbar]");  
const navTogglers = document.querySelectorAll("[data-nav-toggler]");  
const overlay = document.querySelector("[data-overlay]");  
  
const toggleNavbar = function () {  
  navbar.classList.toggle("active");  
  overlay.classList.toggle("active");  
  document.body.classList.toggle("nav-active");  
}  
addEventOnElements(navTogglers, "click", toggleNavbar);  
/**
```

```
* HEADER & BACK TOP BTN
```

```
*/
```

```
const header = document.querySelector("[data-header]");  
const backTopBtn = document.querySelector("[data-back-top-btn]");  
let lastScrollPos = 0;  
const hideHeader = function () {  
    const isScrollBottom = lastScrollPos < window.scrollY;  
    if (isScrollBottom) {  
        header.classList.add("hide");  
    } else {  
        header.classList.remove("hide");  
    }  
    lastScrollPos = window.scrollY;  
}  
window.addEventListener("scroll", function () {  
    if (window.scrollY >= 50) {  
        header.classList.add("active");  
        backTopBtn.classList.add("active");  
        hideHeader();  
    } else {  
        header.classList.remove("active");  
        backTopBtn.classList.remove("active");  
    }  
});  
/**
```

```
* HERO SLIDER
```

```
*/
```

```
const heroSlider = document.querySelector("[data-hero-slider]");  
const heroSliderItems = document.querySelectorAll("[data-hero-slider-item]");  
const heroSliderPrevBtn = document.querySelector("[data-prev-btn]");  
const heroSliderNextBtn = document.querySelector("[data-next-btn]");  
let currentSlidePos = 0;  
let lastActiveSliderItem = heroSliderItems[0];
```

```
const updateSliderPos = function () {
  lastActiveSliderItem.classList.remove("active");
  heroSliderItems[currentSlidePos].classList.add("active");
  lastActiveSliderItem = heroSliderItems[currentSlidePos];
}

const slideNext = function () {
  if (currentSlidePos >= heroSliderItems.length - 1) {
    currentSlidePos = 0;
  } else {
    currentSlidePos++;
  }

  updateSliderPos();
}

heroSliderNextBtn.addEventListener("click", slideNext);

const slidePrev = function () {
  if (currentSlidePos <= 0) {
    currentSlidePos = heroSliderItems.length - 1;
  } else {
    currentSlidePos--;
  }

  updateSliderPos();
}

heroSliderPrevBtn.addEventListener("click", slidePrev);

/**
 * auto slide
 */
let autoSlideInterval;

const autoSlide = function () {
  autoSlideInterval = setInterval(function () {
```



```

slideNext();
}, 7000);
}

addEventListener([heroSliderNextBtn, heroSliderPrevBtn], "mouseover", function ()
{
clearInterval(autoSlideInterval);
});

addEventListener([heroSliderNextBtn, heroSliderPrevBtn], "mouseout", autoSlide);

window.addEventListener("load", autoSlide);

/**
 * PARALLAX EFFECT
 */
const parallaxItems = document.querySelectorAll("[data-parallax-item]");

let x, y;

window.addEventListener("mousemove", function (event) {

x = (event.clientX / window.innerWidth * 10) - 5;
y = (event.clientY / window.innerHeight * 10) - 5;

// reverse the number eg. 20 -> -20, -5 -> 5
x = x - (x * 2);
y = y - (y * 2);

for (let i = 0, len = parallaxItems.length; i < len; i++) {
x = x * Number(parallaxItems[i].dataset.parallaxSpeed);
y = y * Number(parallaxItems[i].dataset.parallaxSpeed);
parallaxItems[i].style.transform = `translate3d(${x}px, ${y}px, 0px)`;
}
});

```

ДОДАТОК Г

Html код сайту

```
<!DOCTYPE html>
<html lang="ua">

<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<!--
- primary meta tags
-->
<title>Ресторан італійської кухні</title>
<meta name="title" content="Ресторан італійської кухні">
<!--
- favicon
-->
<link rel="shortcut icon" href="/favicon.svg"
type="image/svg+xml">
<!--
- google font link
-->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=DM+Sans:wght@400;700&family=Forum&dis
play=swap"
rel="stylesheet">
<!--
- custom css link
-->
<link rel="stylesheet" href="/assets/css/style.css">
<!--
```

```
- preload images
```

```
-->
```

```
<link rel="preload" as="image" href="./assets/images/hero-slider-1.jpg">
```

```
<link rel="preload" as="image" href="./assets/images/hero-slider-2.jpg">
```

```
<link rel="preload" as="image" href="./assets/images/hero-slider-3.jpg">
```

```
</head>
```

```
<body id="top">
```

```
<!--
```

```
- #PRELOADER
```

```
-->
```

```
<div class="preload" data-preload>
```

```
<div class="circle"></div>
```

```
<p class="text">Grilli</p>
```

```
</div>
```

```
<!--
```

```
- #TOP BAR
```

```
-->
```

```
<div class="topbar">
```

```
<div class="container">
```

```
<address class="topbar-item">
```

```
<div class="icon">
```

```
<ion-icon name="location-outline" aria-hidden="true"></ion-icon>
```

```
</div>
```

```
<span class="span">
```

```
вулиця Кіото, 19, Київ
```

```
</span>
```

```
</address>
```

```
<div class="separator"></div>
```

```
<div class="topbar-item item-2">
```

```
<div class="icon">
```

```
<ion-icon name="time-outline" aria-hidden="true"></ion-icon>
```

```
</div>
```

```

    <span class="span">Щодня : з 8.00 до 22:00</span>
  </div>
  <a href="tel:+11234567890" class="topbar-item link">
    <div class="icon">
      <ion-icon name="call-outline" aria-hidden="true"></ion-icon>
    </div>
    <span class="span">+380 994 629 123</span>
  </a>
  <div class="separator"></div>
  <a href="mailto:booking@restaurant.com" class="topbar-item link">
    <div class="icon">
      <ion-icon name="mail-outline" aria-hidden="true"></ion-icon>
    </div>
    <span class="span">booking@restaurant.com</span>
  </a>
</div>
</div>
<!--
- #HEADER
-->
<header class="header" data-header>
  <div class="container">
    <a href="#" class="logo">
      
    </a>
    <nav class="navbar" data-navbar>
      <button class="close-btn" aria-label="close menu"
        data-nav-toggler>
        <ion-icon name="close-outline" aria-hidden="true"></ion-icon>
      </button>
      <a href="#" class="logo">
        

```

```
</a>
<ul class="navbar-list">
  <li class="navbar-item">
    <a href="#home" class="navbar-link hover-underline active">
      <div class="separator"></div>
      <span class="span">Головна</span>
    </a>
  </li>
  <li class="navbar-item">
    <a href="#menu" class="navbar-link hover-underline">
      <div class="separator"></div>
      <span class="span">Меню</span>
    </a>
  </li>
  <li class="navbar-item">
    <a href="#about" class="navbar-link hover-underline">
      <div class="separator"></div>
      <span class="span">Про Нас</span>
    </a>
  </li>
  <li class="navbar-item">
    <a href="#" class="navbar-link hover-underline">
      <div class="separator"></div>
      <span class="span">Наші Кухарі</span>
    </a>
  </li>
  <li class="navbar-item">
    <a href="#" class="navbar-link hover-underline">
      <div class="separator"></div>
      <span class="span">Контакти</span>
    </a>
  </li>
</ul>
<div class="text-center">
```

```

<p class="headline-1 navbar-title">Відвідайте Нас</p>
<address class="body-4">
  вулиця Кіото, 19, Київ
</address>
<p class="body-4 navbar-text">Відкрито: 9:00 - 22:00pm</p>
<a href="mailto:booking@grilli.com" class="body-4 sidebar-
link">booking@grilli.com</a>
<div class="separator"></div>
<p class="contact-label">Бронювання</p>
<a href="tel:+88123123456" class="body-1 contact-number hover-underline">
+380 994 629 123
</a>
</div>
</nav>
<a href="#" class="btn btn-secondary">
<span class="text text-1">Знайти стіл</span>
<span class="text text-2" aria-hidden="true">Знайти стіл</span>
</a>
<button class="nav-open-btn" aria-label="open menu"
data-nav-toggler>
<span class="line line-1"></span>
<span class="line line-2"></span>
<span class="line line-3"></span>
</button>
<div class="overlay" data-nav-toggler data-overlay></div>
</div>
</header>
<main>
<article>
<!--
- #HERO
-->
<section class="hero text-center" aria-label="home"
id="home">

```

```

<ul class="hero-slider" data-hero-slider>
  <li class="slider-item active" data-hero-slider-item>
    <div class="slider-bg">
      
    </div>
    <p class="label-2 section-subtitle slider-reveal">Традиції та Гігієна</p>
    <h1 class="display-1 hero-title slider-reveal">
      З любов'ю до<br>
      смачної їжі
    </h1>
    <p class="body-2 hero-text slider-reveal">
      Приходьте з сім'єю та відчуйте радість від
      апетитної їжі
    </p>
    <a href="#" class="btn btn-primary slider-reveal">
      <span class="text text-1">Переглянути Наше Меню</span>
      <span class="text text-2" aria-hidden="true">Переглянути Наше
      Меню</span>
    </a>
  </li>
  <li class="slider-item" data-hero-slider-item>
    <div class="slider-bg">
      
    </div>
    <p class="label-2 section-subtitle slider-reveal">чудовий досвід</p>
    <h1 class="display-1 hero-title slider-reveal">
      Аромати, натхненні<br>
      порами року
    </h1>
    <p class="body-2 hero-text slider-reveal">

```

Приходьте з сім'єю та відчуйте радість від
апетитної їжі

</p>

Переглянути Наше Меню

Переглянути наше меню

<li class="slider-item" data-hero-slider-item>

<div class="slider-bg">

</div>

<p class="label-2 section-subtitle slider-reveal">дивовижні та смачні</p>

<h1 class="display-1 hero-title slider-reveal">

Де кожен смак

розповідає історію

</h1>

<p class="body-2 hero-text slider-reveal">

Приходьте з сім'єю та відчуйте радість від
апетитної їжі

</p>

Переглянути Наше Меню

Переглянути Наше

Меню

<button class="slider-btn prev" aria-label="slide to previous"

data-prev-btn>

<ion-icon name="chevron-back"></ion-icon>

</button>


```

<button class="slider-btn next" aria-label="slide to next"
data-next-btn>
  <ion-icon name="chevron-forward"></ion-icon>
</button>
<a href="#" class="hero-btn has-after">
  
  <span class="label-2 text-center span">Забронюйте Столик</span>
</a>
</section>
<!--
- #SERVICE
-->
<section class="section service bg-black-10 text-center"
aria-label="service">
  <div class="container">
    <p class="section-subtitle label-2">Аромати для Королів</p>
    <h2 class="headline-1 section-title">Ми пропонуємо найвищий рівень</h2>
    <p class="section-text">
      Lorem Ipsum is simply dummy text of the printing
      and typesetting industry lorem
      Ipsum has been the industrys
      standard dummy text ever.
    </p>
    <ul class="grid-list">
      <li>
        <div class="service-card">
          <a href="#" class="has-before hover:shine">
            <figure class="card-banner img-holder" style="--width: 285; --height: 336;">
              
            </figure>

```

```

</a>
<div class="card-content">
  <h3 class="title-4 card-title">
    <a href="#">Сніданок</a>
  </h3>
  <a href="#" class="btn-text hover-underline label-2">Переглянути
Меню</a>
</div>
</div>
</li>
<li>
  <div class="service-card">
    <a href="#" class="has-before hover:shine">
      <figure class="card-banner img-holder" style="--width: 285; --height: 336;">
        
      </figure>
    </a>
    <div class="card-content">
      <h3 class="title-4 card-title">
        <a href="#">Закуски</a>
      </h3>
      <a href="#" class="btn-text hover-underline label-2">Переглянути
Меню</a>
    </div>
  </div>
</li>
<li>
  <div class="service-card">
    <a href="#" class="has-before hover:shine">
      <figure class="card-banner img-holder" style="--width: 285; --height: 336;">
        
</figure>
</a>
<div class="card-content">
<h3 class="title-4 card-title">
<a href="#">Напої</a>
</h3>
<a href="#" class="btn-text hover-underline label-2">Переглянути
Меню</a>
</div>
</div>
</li>
</ul>


</div>
</section>
<!--
- #ABOUT
-->
<section class="section about text-center"
aria-labelledby="about-label" id="about">
<div class="container">
<div class="about-content">
<p class="label-2 section-subtitle" id="about-label">Наша Історія</p>
<h2 class="headline-1 section-title">Every Fla vor Tells a Story</h2>
<p class="section-text">
Lorem Ipsum is simply dummy text of the printingand

```

typesetting industry lorem
 Ipsum has been the industrys
 standard dummy text ever since
 the when an unknown printer
 took a galley of type and scrambled
 it to make a type specimen
 book It has survived not only
 five centuries, but also the
 leap into.

```
</p>
```

```
<div class="contact-label">Забронювати через дзвінок</div>
```

```
<a href="tel:+804001234567" class="body-1 contact-number hover-underline">+380 994 629 123</a>
```

```
<a href="#" class="btn btn-primary">
```

```
<span class="text text-1">Читати більше</span>
```

```
<span class="text text-2" aria-hidden="true">Читати більше</span>
```

```
</a>
```

```
</div>
```

```
<figure class="about-banner">
```

```

```

```
<div class="abs-img abs-img-1 has-before"
```

```
data-parallax-item data-parallax-speed="1.75">
```

```

```

```
</div>
```

```
<div class="abs-img abs-img-2 has-before">
```

```

```

```
</div>
```

```

</figure>

</div>
</section>
<!--
- #SPECIAL DISH
-->
<section class="special-dish text-center"
aria-labelledby="dish-label">
<div class="special-dish-banner">

</div>
<div class="special-dish-content bg-black-10">
<div class="container">

<p class="section-subtitle label-2">Фірмова страва</p>
<h2 class="headline-1 section-title">Тортелліні з лобстерами</h2>
<p class="section-text">
Lorem Ipsum is simply dummy text of the printing and
typesetting industry lorem
Ipsum has been the industry's
standard dummy text ever since
the when an unknown printer
took a galley of type.
</p>
<div class="wrapper">
<del class="del body-3">$40.00</del>
<span class="span body-1">$20.00</span>

```

```

</div>
<a href="#" class="btn btn-primary">
  <span class="text text-1">Переглянути Все Меню</span>
  <span class="text text-2" aria-hidden="true">Переглянути Все Меню</span>
</a>
</div>
</div>


</section>
<!--
- #MENU
-->
<section class="section menu" aria-label="menu-label"
id="menu">
  <div class="container">
    <p class="section-subtitle text-center label-2">Спеціальна добірка</p>
    <h2 class="headline-1 section-title text-center">Вишукане меню</h2>
    <ul class="grid-list">
      <li>
        <div class="menu-card hover:card">
          <figure class="card-banner img-holder" style="--width: 100; --height: 100;">
            
          </figure>
          <div>
            <div class="title-wrapper">
              <h3 class="title-3">
                <a href="#" class="card-title">Грецький салат</a>

```

```

</h3>
<span class="badge label-1">Сезонний</span>
<span class="span title-2">$25.50</span>
</div>
<p class="card-text label-1">
Помідори, зелений болгарський перець, нарізані
огірки, цибуля, оливки
та бринза.
</p>
</div>
</div>
</li>
<li>
<div class="menu-card hover:card">
<figure class="card-banner img-holder" style="--width: 100%; --height: 100%;">

</figure>
<div>
<div class="title-wrapper">
<h3 class="title-3">
<a href="#" class="card-title">Лазанья</a>
</h3>
<span class="span title-2">$40.00</span>
</div>
<p class="card-text label-1">
Овочі, сири, м'ясний фарш, томатний соус,
приправи та спеції
</p>
</div>
</div>
</li>
</li>

```

```

<div class="menu-card hover:card">
  <figure class="card-banner img-holder" style="--width: 100; --height: 100;">
    
  </figure>
</div>
<div class="title-wrapper">
  <h3 class="title-3">
    <a href="#" class="card-title">Мускатний гарбуз</a>
  </h3>
  <span class="span title-2">$10.00</span>
</div>
<p class="card-text label-1">
  Typesetting industry lorem Lorem Ipsum is
  simply dummy text of
  the priand.
</p>
</div>
</div>
</li>
<li>
  <div class="menu-card hover:card">
    <figure class="card-banner img-holder" style="--width: 100; --height: 100;">
      
    </figure>
  <div class="title-wrapper">
    <h3 class="title-3">
      <a href="#" class="card-title">Токусен Вагю</a>
    </h3>

```



```

<span class="badge label-1">Новинка</span>
<span class="span title-2">$39.00</span>
</div>
<p class="card-text label-1">
  Овочі, сири, м'ясний фарш, томатний соус,
  приправи та спеції.
</p>
</div>
</div>
</li>
<li>
  <div class="menu-card hover:card">
    <figure class="card-banner img-holder" style="--width: 100; --height: 100;">
      
    </figure>
    <div>
      <div class="title-wrapper">
        <h3 class="title-3">
          <a href="#" class="card-title">Олівас Релленас</a>
        </h3>
        <span class="span title-2">$25.00</span>
      </div>
      <p class="card-text label-1">
        Авокадо з крабовим м'ясом, червоною цибулею,
        крабовим салатом, фаршированим
        червоним болгарським
        перцем і зеленим болгарським
        перцем.
      </p>
    </div>
  </div>
</li>

```

```

</li>
<div class="menu-card hover:card">
<figure class="card-banner img-holder" style="--width: 100%; --height: 100%;">

</figure>
<div>
<div class="title-wrapper">
<h3 class="title-3">
<a href="#" class="card-title">Риба Опу</a>
</h3>
<span class="span title-2">$49.00</span>
</div>
<p class="card-text label-1">
Овочі, сири, м'ясний фарш, томатний соус,
приправи та спеції
</p>
</div>
</div>
</li>
</ul>
<p class="menu-text text-center">
Взимку щодня від <span class="span">7:00 </span> до <span
class="span">21:00 </span>
</p>
<a href="#" class="btn btn-primary">
<span class="text text-1">Переглянути Все Меню</span>
<span class="text text-2" aria-hidden="true">Переглянути Все Меню</span>
</a>


</div>
</section>
<!--
- #TESTIMONIALS
-->
<section class="section testi text-center has-bg-image"
style="background-image: url('./assets/images/testimonial-bg.jpg')"
aria-label="testimonials">
<div class="container">
<div class="quote"></div>
<p class="headline-2 testi-text">
Я хотів би подякувати вам за запрошення на
чудову вечерю вчора ввечері.
Їжа була надзвичайною.
</p>
<div class="wrapper">
<div class="separator"></div>
<div class="separator"></div>
<div class="separator"></div>
</div>
<div class="profile">

<p class="label-2 profile-name">Максим Биков </p>
</div>
</div>
</section>
<!--
- #RESERVATION
-->
<section class="reservation">

```

```

<div class="container">
  <div class="form reservation-form bg-black-10">
    <form action="" class="form-left">
      <h2 class="headline-1 text-center">Бронювання онлайн</h2>
      <p class="form-text text-center">
        Забронювати <a href="tel:+88123123456"
        class="link">+380 994 629 123</a> або заповніть форму
      </p>
      <div class="input-wrapper">
        <input type="text" name="name" placeholder="Ваше Ім'я"
        autocomplete="off" class="input-field">
        <input type="tel" name="phone" placeholder="Номер Телефону"
        autocomplete="off" class="input-field">
      </div>
      <div class="input-wrapper">
        <div class="icon-wrapper">
          <ion-icon name="person-outline" aria-hidden="true"></ion-icon>
          <select name="person" class="input-field">
            <option value="1-person">1 Персона</option>
            <option value="2-person">2 Персони</option>
            <option value="3-person">3 Персони</option>
            <option value="4-person">4 Персони</option>
            <option value="5-person">5 Персон</option>
            <option value="6-person">6 Персон</option>
            <option value="7-person">7 Персон</option>
          </select>
          <ion-icon name="chevron-down" aria-hidden="true"></ion-icon>
        </div>
        <div class="icon-wrapper">
          <ion-icon name="calendar-clear-outline" aria-hidden="true"></ion-icon>
          <input type="date" name="reservation-date"
          class="input-field">
          <ion-icon name="chevron-down" aria-hidden="true"></ion-icon>

```

замовлення

```
</div>
<div class="icon-wrapper">
  <ion-icon name="time-outline" aria-hidden="true"></ion-icon>
  <select name="person" class="input-field">
    <option value="08:00am">08 : 00</option>
    <option value="09:00am">09 : 00</option>
    <option value="10:00am">10 : 00</option>
    <option value="11:00am">11 : 00</option>
    <option value="12:00am">12 : 00</option>
    <option value="01:00pm">13 : 00</option>
    <option value="02:00pm">14 : 00</option>
    <option value="03:00pm">15 : 00</option>
    <option value="04:00pm">16 : 00</option>
    <option value="05:00pm">17 : 00</option>
    <option value="06:00pm">18 : 00</option>
    <option value="07:00pm">19 : 00 </option>
    <option value="08:00pm">20 : 00</option>
    <option value="09:00pm">21 : 00</option>
  </select>
  <ion-icon name="chevron-down" aria-hidden="true"></ion-icon>
</div>
</div>
<textarea name="message" placeholder="Коментар"
autocomplete="off" class="input-field"></textarea>
<button type="submit" class="btn btn-secondary">
  <span class="text text-1">Забронювати столик</span>
  <span class="text text-2" aria-hidden="true">Забронювати столик</span>
</button>
</form>
<div class="form-right text-center" style="background-image:
url('./assets/images/form-pattern.png')">
  <h2 class="headline-1 text-center">Зв'яжіться з нами</h2>
  <p class="contact-label">Забронювати</p>
```

[+380 994 629 123](tel:+88123123456)

<div class="separator"></div>

<p class="contact-label">Місцезнаходжен</p>

<address class="body-4">

вулиця Кіото, 19, Київ

</address>

<p class="contact-label">Ланч</p>

<p class="body-4">

3 Понеділка по Неділю

 11:00 - 14:30

</p>

<p class="contact-label">Вечеря</p>

<p class="body-4">

3 Понеділка по Неділю

 17:00 - 22:00

</p>

</div>

</div>

</div>

</section>

<!--

- #FEATURES

-->

<section class="section features text-center"

aria-label="features">

<div class="container">

<p class="section-subtitle label-2">Чому Обирають Нас</p>

<h2 class="headline-1 section-title">Наша Перевага</h2>

<ul class="grid-list">

<li class="feature-item">

<div class="feature-card">

<div class="card-icon">

</div>
<h3 class="title-2 card-title">Hygienic Food</h3>
<p class="label-1 card-text">Lorem Ipsum is simply dummy
printing and typesetting.</p>
</div>

<li class="feature-item">
<div class="feature-card">
<div class="card-icon">

</div>
<h3 class="title-2 card-title">Fresh Environment</h3>
<p class="label-1 card-text">Lorem Ipsum is simply dummy
printing and typesetting.</p>
</div>

<li class="feature-item">
<div class="feature-card">
<div class="card-icon">

</div>
<h3 class="title-2 card-title">Досвідчені Кухарі</h3>
<p class="label-1 card-text">Lorem Ipsum is simply dummy
printing and typesetting.</p>
</div>

<li class="feature-item">
<div class="feature-card">

```

```

<div class="card-icon">

</div>
<h3 class="title-2 card-title">Події та Вечірки</h3>
<p class="label-1 card-text">Lorem Ipsum is simply dummy
printing and typesetting.</p>
</div>

</div>
</section>
<!--
- #EVENT
-->
<section class="section event bg-black-10"
aria-label="event">
<div class="container">
 <p class="section-subtitle label-2 text-center">Останні оновлення</p>
 <h2 class="section-title headline-1 text-center">Найближча подія</h2>
 <ul class="grid-list">

 <div class="event-card has-before hover:shine">
 <div class="card-banner img-holder" style="--width: 350; --height: 450;">


```



```

class="img-cover">
<time class="publish-date label-2" datetime="2023-03-
11">11/03/2023</time>
</div>
<div class="card-content">
<p class="card-subtitle label-2 text-center">Їжа, Смак</p>
<h3 class="card-title title-2 text-center">
Смак настільки хороший, що хочеться їсти очима.
</h3>
</div>
</div>

<div class="event-card has-before hover:shine">
<div class="card-banner img-holder" style="--width: 350; --height: 450;">

<time class="publish-date label-2" datetime="2023-03-
12">12/03/2023</time>
</div>
<div class="card-content">
<p class="card-subtitle label-2 text-center">Здорова Їжа</p>
<h3 class="card-title title-2 text-center">
Смак настільки хороший, що хочеться їсти очима.
</h3>
</div>
</div>

<div class="event-card has-before hover:shine">
<div class="card-banner img-holder" style="--width: 350; --height: 450;">

<time class="publish-date label-2" datetime="2023-03-
13">13/03/2023</time>
</div>
<div class="card-content">
<p class="card-subtitle label-2 text-center">Рецепти</p>
<h3 class="card-title title-2 text-center">
Рецепти
</h3>
</div>
</div>

Переглянути Наш Блог
Переглянути Наш Блог

</div>
</section>
</article>
</main>
<!--
- #FOOTER
-->
<footer class="footer section has-bg-image text-center"
style="background-image: url('./assets/images/footer-bg.jpg')">
<div class="container">
<div class="footer-top grid-list">
<div class="footer-brand has-before has-after">


```

&lt;/a&gt;

&lt;address class="body-4"&gt;

вулиця Кіото, 19, Київ

&lt;/address&gt;

&lt;a href="mailto:booking@grilli.com" class="body-4 contact-link"&gt;booking@grilli.com&lt;/a&gt;

<a href="tel:+88123123456" class="body-4 contact-link">Бронювання : +380 994  
629 123</a>

&lt;p class="body-4"&gt;

Відкрито : з 08:00 до 22:00

&lt;/p&gt;

&lt;div class="wrapper"&gt;

&lt;div class="separator"&gt;&lt;/div&gt;

&lt;div class="separator"&gt;&lt;/div&gt;

&lt;div class="separator"&gt;&lt;/div&gt;

&lt;/div&gt;

&lt;p class="title-1"&gt;Отримуйте новини та пропозиції&lt;/p&gt;

&lt;p class="label-1"&gt;

Підпишіться на нас та отримайте

&lt;span

class="span"&gt;25% ЗНИЖКИ.&lt;/span&gt;

&lt;/p&gt;

&lt;form action="" class="input-wrapper"&gt;

&lt;div class="icon-wrapper"&gt;

&lt;ion-icon name="mail-outline" aria-hidden="true"&gt;&lt;/ion-icon&gt;

&lt;input type="email" name="email\_address"

placeholder="Your email" autocomplete="off"

class="input-field"&gt;

&lt;/div&gt;

&lt;button type="submit" class="btn btn-secondary"&gt;

&lt;span class="text text-1"&gt;Підписатися&lt;/span&gt;

&lt;span class="text text-2" aria-hidden="true"&gt;Підписатися&lt;/span&gt;

&lt;/button&gt;

&lt;/form&gt;

```
</div>
<ul class="footer-list">

 Головна

 Меню

 Про нас

 Наші Кухарі

 Контакти

<ul class="footer-list">

 Facebook

 Instagram

 Twitter

 Youtube

 Google Map


```

```

</div>
<div class="footer-bottom">
</div>
</div>
</footer>
<!--
- #BACK TO TOP
-->
<a href="#top" class="back-top-btn active"
aria-label="back to top" data-back-top-btn>
 <ion-icon name="chevron-up" aria-hidden="true"></ion-icon>

<!--
- custom js link
-->
<script src="./assets/js/script.js"></script>
<!--
- ionic link
-->
<script type="module"
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>
<script nomodule
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js"></script>
</body>
</html>
```