

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента Варежкіна Іллі Олександровича
(ПІБ)

академічної групи 122-18-2
(шифр)

спеціальності 122 Комп'ютерні науки
(код і назва спеціальності)

освітньої програми Комп'ютерні науки
(назва освітньої програми)

на тему: Розробка інформаційного веб-додатку для італійського ресторану

| Керівники | Прізвище, ініціали | Оцінка за шкалою | | Підпис |
|------------------------|-----------------------------|------------------|---------------|--------|
| | | рейтинговою | інституційною | |
| кваліфікаційної роботи | <i>Проф. Мещеряков Л.І.</i> | | | |
| розділів: | | | | |
| спеціальний | <i>Проф. Мещеряков Л.І.</i> | | | |
| економічний | <i>доц. Касьяненко Л.В.</i> | | | |
| | | | | |
| | | | | |
| Рецензент | | | | |
| Нормоконтролер | <i>доц. Гуліна І.Г.</i> | | | |

Дніпро
2022

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

завідувач кафедри
програмного забезпечення комп'ютерних систем

(повна назва)

І.М. Удовик

(підпис)

(прізвище, ініціали)

« » _____ 2022 року

ЗАВДАННЯ

на кваліфікаційну роботу

бакалавра

(назва освітньо-кваліфікаційного рівня)

студента 122-18-2
(група)

Варежкіна І.О.

(прізвище та ініціали)

тема кваліфікаційної роботи

Розробка інформаційного веб-додатку для
італійського ресторану.

затверджена наказом ректора НТУ «ДП» від

№

| Розділ | Зміст виконання | Термін виконання |
|-------------|--|------------------|
| Спеціальний | <i>На основі матеріалів проєктно-технологічної практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми</i> | 13.05.2022 р. |
| Економічний | <i>Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки</i> | 27.05.2022 р. |

Завдання видав

Проф. Мещеряков Л.І.

(підпис)

(посада, прізвище, ініціали)

Завдання прийняв до виконання

Варежкін І.О.

(підпис)

(прізвище, ініціали)

Дата видачі завдання: 14.01.2022 р.

Термін подання кваліфікаційної роботи до ЕК: 13.06.2022 р.

РЕФЕРАТ

Пояснювальна записка: ___ с., ___ рис., ___ табл., ___ дод., ___ джерел.

Об'єкт розробки: інформаційний веб-додаток для італійського ресторану.

Мета кваліфікаційної роботи: створення інформаційного веб-додатку для охопту нових клієнтів, збільшення доходу та інших показників ресторану.

У вступі розглядається аналіз проблеми, детально розкривається мета кваліфікаційної роботи та область її застосування, обґрунтовується актуальність теми та пояснюється суть завдання.

У першому розділі аналізується предметна галузь, визначається актуальність завдання та призначення розробки, обґрунтовується суть завдання, визначаються вимоги до програмних засобів, програмної реалізації та технологій.

У другому розділі аналізуються наявні рішення, обираються платформи та інструменти для розробки, також виконується аналіз продукту, проектування його каркасу, створення інтерфейсу користувача, створення інтерактивного прототипу та дизайн-системи, впровадження веб-додатку.

В економічному розділі визначається трудомісткість розробленого інформаційного веб-додатку, проведений підрахунок вартості роботи по створенню додатку в годинах.

Практичне значення полягає у створенні веб-додатку, який дає змогу користувачу ознайомитись з рестораном, а саме, дізнатись про його історію виникнення, ознайомитись з наявним меню та дає змогу забронювати столик онлайн.

Актуальність даного продукту визначається великим попитом на схожі розробки, що дають потужний прорив у маркетинговій стратегії та додатковий дохід.

Список ключових слів: ВЕБ-ДОДАТОК, РЕСТОРАН, САЙТ, ІНТЕРФЕЙС КОРИСТУВАЧА.

ABSTRACT

Explanatory note: ___ pp., ___ fig., ___ table, ___ appendix, ___ sources.

Development object: informational web application for an Italian restaurant.

The purpose of the qualification work: to create an informational web application to reach new customers, increase revenue and other indicators of the restaurant.

The introduction considers the analysis of the problem, reveals in detail the purpose of the qualification work and its scope, substantiates the relevance of the topic and explains the essence of the task.

The first section analyzes the subject area, determines the relevance of the task and the purpose of development, substantiates the essence of the task, determines the requirements for software, software implementation and technology.

The second section analyzes the existing solutions, selects platforms and tools for development, also analyzes the product, designing its framework, creating a user interface, creating an interactive prototype and design system, implementing a web application.

The economic section determines the complexity of the developed information web application, the calculation of the cost of work to create an application in hours.

The practical value is to create a web application that allows the user to get acquainted with the restaurant, namely, to learn about its history, to read the available menu and allows you to book a table online.

The relevance of this product is determined by the high demand for similar developments, which give a powerful breakthrough in marketing strategy and additional income.

List of keywords: WEB APP, RESTAURANT, WEBSITE, USER INTERFACE.

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

UX — User Experience;

UI — User Interface;

HTML — HyperText Markup Language;

CSS — Cascading Style Sheets;

JS — JavaScript;

BEM — Block Element Modifier;

ЗМІСТ

| | |
|--|----|
| РЕФЕРАТ..... | 3 |
| ABSTRACT..... | 4 |
| СПИСОК УМОВНИХ ПОЗНАЧЕНЬ..... | 5 |
| ВСУП..... | 8 |
| РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧ..... | 9 |
| 1.1. Загальні відомості з предметної галузі..... | 9 |
| 1.2. Призначення розробки та галузь застосування..... | 14 |
| 1.3. Підстави для розробки..... | 14 |
| 1.4. Постановка завдання..... | 14 |
| 1.5. Вимоги до програми або програмного вибору..... | 15 |
| 1.5.1. Вимоги до функціональних характеристик..... | 15 |
| 1.5.2. Вимоги до інформаційної безпеки..... | 15 |
| 1.5.3. Вимоги до складу та параметрів технічних засобів..... | 16 |
| 1.5.4. Вимоги до інформаційної та програмної сумісності..... | 16 |
| РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ..... | 17 |
| 2.1. Функціональне призначення програми..... | 17 |
| 2.2. Опис застосованих математичних методів..... | 17 |
| 2.3. Опис використаних технології та мов програмування..... | 17 |
| 2.4. Опис структури системи та алгоритмів її функціонування..... | 21 |
| 2.5. Обґрунтування та організація вхідних та вихідних даних програми..... | 27 |
| 2.6. Опис розробленої системи..... | 27 |
| 2.6.1. Використані технічні засоби..... | 27 |
| 2.6.2. Використані програмні засоби..... | 29 |
| 2.6.3. Виклик та завантаження програми..... | 31 |
| 2.6.4. Опис інтерфейсу користувача..... | 32 |
| РОЗДІЛ 3 ЕКОНОМІЧНИЙ РОЗДІЛ..... | 40 |
| 3.1. Розрахунок трудомісткості та вартості розробки програмного продукту.... | 40 |

| | |
|---|----|
| 3.2. Розрахунок витрат на створення програми..... | 44 |
| ВИСНОВОК..... | 46 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 47 |
| ДОДАТОК А. КОД ПРОГРАМИ..... | 49 |
| ДОДАТОК Б. ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ..... | 76 |
| ДОДАТОК В. ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ..... | 77 |

ВСТУП

Зараз дуже динамічне життя серед молодих людей, та через великі амбіції не вистачає часу і сил на звичайні речі, навіть на приготування їжі.

Тому похід у кафе або ресторан, це гарне рішення. Але ж подібних закладів багато і незрозуміло, де яка їжа, куди краще піти? Саме для цього є чудове рішення, це веб-додаток ресторану з усією інформацією про меню та про сам заклад, а також з можливістю забронювати столик онлайн. Ще додаток має змогу повідомляти про режим роботи, новини та містити у собі реальні відгуки відвідувачів.

Власникам ресторанного бізнесу зараз дуже актуально мати власний сайт ресторану, тому що з'являється можливість робити зворотній зв'язок з клієнтами, покращується статусність та популярність закладу, є змога швидко поширювати інформацію щодо графіку роботи, зміни у меню і т. д., та саме головне, це залучення нових потенційних відвідувачів. Також до веб-додатку можна підключити систему доставки їжі, що дає змогу примножити дохід ресторану. А ще до додатку можливо прив'язати "Google Analytics", це сервіс який створює детальну статистику відвідувачів, завдяки цьому можна проводити маркетингові дослідження. Усе це дуже сильно розвиває ваш бізнес.

Тому проблема, яка розглянута в даній кваліфікаційній роботі є актуальною.

Мета роботи — розробити веб-додаток для італійського ресторану.

Основне завдання даної роботи це:

- сформулювати загальне уявлення про продукт, його аудиторію та задачі;
- визначити конкретні механіки для досягнення цілей;
- розробити концепцію;
- обрати мову програмування;
- розробити веб-додаток для італійського ресторану.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Загальні відомості з предметної галузі

Останнім часом багато людей у котрих є свій бізнес зустрічаються з інформацією про комерційні веб-додатки. Зараз в інтернеті існує немала кількість комерційного підходу, а саме можна продавати або рекламувати свої послуги, товари, тощо. Основне завдання комерційного веб-додатку являється здобування прибутку.

Раніше багато веб-додатків не мали змогу прямого прибутку, та вони сприяли зростанню прибутку через звичайні засоби. Головне призначення та мета існування таких веб-додатків є привертання уваги клієнтів до комерційної діяльності. Тепер інтернет дає перспективи електронної комерції.

Зараз багато власників великого та навіть малого бізнесу продають свої послуги та товари через інтернет. Комерційні веб-додатки, котрі займаються продажем товарів мають назву інтернет-магазин.

Також існують корпоративні веб-додатки. Це додатки фірм, які надають певні послуги, наприклад : ресторани, кафе, адвокати та інші юридичні напрями, рекламодавці з СММ та таргету, фірми по створенню сайтів та мобільних додатків, фітнес, йога і т. д.

Головна ціль таких комерційних веб-додатків — отримання прибутку завдяки надання послуг людям, а не через продаж товарів. Притому самі послуги залишаються поза межами інтернету. Веб-додаток є головним інструментом реклами, який буде у відвідувача уяву про фірму, яка дуже приваблює та сильно запам'ятовується. Мета власників веб-додатку, це переконати читачів, що вам дуже цікаві і потрібні саме їх фірма.

Комерційні веб-додатки є одним за напрямків в галузі створення веб-додатків. Головна мета цих додатків — це отримання вигоди, котра виражена у збільшені доходів компанії та в зростанні інших комерційних показників. Мета та завдання комерційних веб-додатків багато відрізняються від мети та завдань

інших веб-додатків, саме тому розробка комерційного веб-додатку дуже відрізняється від інших додатків.

Головне завдання комерційного додатку — це отримання прибутку. Продаючи послуги або товари онлайн, це вид комерційного веб-додатку як корпоративний додаток який добре виконує саме це завдання. Також ще одна вигода, отримання якої стає метою розробки комерційного веб-додатку — залучення нових потенційних клієнтів.

Перед тим, як працювати з веб-додатком, треба побудувати чітке уявлення про його структуру. Це впливає на розподіл ресурсів у пошукових системах (наприклад Google), та на його сприйняття користувачами. Коли дізнаєшся про грамотну структуру, тоді стає зрозуміло, чому схожі між собою веб-додатки за параметрами мають зовсім різні показники.

Що взагалі таке структура веб-додатку? Це порядок розташування його сторінок, тобто, наприклад, категорії каталогу товарів. Це як план, котрий будує логічний зв'язок між сторінками. Важливо розуміти, що є зовнішня та внутрішня структура веб-додатку. Зовнішня структура розуміє собою макет сторінки із зазначенням розташування на ній блоків. Внутрішня структура розуміє собою відображення категорій.

Якщо у вас є семантичне ядро і ви точно розумієте, який матеріал повинен бути на сайті, структура ресурсу має бути у вигляді діаграми. Іншими словами – ієрархія, логічний і послідовний ланцюжок побудови та представлення інформації.

Схема може бути різного виду, це залежить від сайту. Тоді обирають найбільш відповідний тип структури:

- Гібридна схема — виключає комбінацію кількох типів структур, що дає змогу користувачам швидше знаходити інформацію.
- Деревоподібна схема — є головна сторінка, та від неї вже йдуть різні категорії, картки товарів тощо. Це поширена структура інтернет магазину.

- Алфавітна організація — розташує інформацію в алфавітному порядку. Зручна для клієнта, котрий точно знає назву товару.
- Тематична схема — зручна навігація для користувачів, коли за темами розташована інформація.

Якщо структура веб-додатку ієрархічна, то вона допомагає користувачам швидко знайти товар, який їх цікавить, не витрачаючи не це зайвого часу.

Веб-додаток – це є сукупністю пов’язаних веб-сторінок, кожна з яких побудується з комп’ютерного коду і підтримується разом під унікальною доменною назвою.

Веб-сторінка – це як сторінка в книжці. Сторінку можна переглядати та взаємодіяти із нею за допомогою веб-браузера (наприклад, Opera, Internet Explorer, Chrome тощо), доступ до якого можна отримати з поїскової системи, з посилань на домашню сторінку вебдодатку. Також веб-додаток – це кількість запитів та відповідей. Щоб отримати доступ до веб-додатку, треба мати доступ до його IP. IP дозволяє користувачам розрізняти буквальні мільярди підключених пристроїв, що складають Інтернет, а також дозволяє цим пристроям точно обмінюватися цифровою інформацією між собою.

При розробці веб-додатку знати необхідні функції, загальний вигляд та тему, параметри навігації по додатку, кількість вмісту та макет інтерфейсу.

Також допомагають оптимізувати дизайн сторінки — карта додатку, а також гарантувати, що всі компоненти працюють разом найбільш ефективно. Оскільки карта додатку може бути простою, як блок-схема, для її створення не потрібні спеціальні навички.

Front-end відрізняється від back-end але не дуже. Чесно кажучи, мови програмування - це майже тільки набори правил, призначених для інструктажу комп’ютерів виконувати певні завдання, більшість мов використовуються для більшості завдань. Коли сайт завантажується в браузер з сервера, то певні дії на стороні клієнта виконуються всередині самого браузера, що дає сторінці нормально функціонувати без потреби підтримувати постійний зв’язок із сервером. Front-end керує всіма запитам та діями користувачів у обмеженому

локальному просторі. Хоча ранні веб-додатки - як правило, що складаються з відформатованого тексту з випадковими GIF-файлами, які вимагали дуже мало у вигляді вдосконаленої інтерфейсної розробки. Тож сьогодні веб-додатки набагато складніші. Мови програмування з Front-end повинні мати можливість відтворювати відео, завантажувати дані з інших сторінок, включати функції ховеру тощо. Для виконання цих завдань, використовуються кілька Front-end мов програмування. Технологічна гілка прогресує зі швидкістю світла.

Прогрес технології розвивається швидше, ніж будь-коли. Інтернет – це найважливіше джерело для пошуку будь-якої інформації. Корінням, котрі дають інформацію в Інтернеті, є веб-додатки. Також існує безліч мов програмування, які використовуються для розробки.

Мова програмування – це мова для запису комп'ютерних систем. Те, що люди записують програмному забезпеченню за допомогою мови програмування, може діяти на те, як виглядала певним чином веб-сторінка, або на те, як об'єкт рухався на сторінці. Кожна веб-сторінка вибудовується з використанням послідовності окремих інструкцій, одна за одною. Браузер є важливим гравцем у перекладі коду на те, що можна бачити на своїх екранах і навіть взаємодіяти. Коли відкривається веб-сторінка, браузер отримує HTML та інші мови програмування, що беруть участь, і інтерпретує їх.

HTML і CSS не є мовами програмування; це інформація про структуру сторінки та стиль. Без знання базових речей HTML та CSS, переходити на інші мови з фронтенду не має сенсу. Бо ці базові речі знаходяться у кожній передовій частині веб-сторінки.

У 90-х років минулого століття HTML була єдиною мовою, доступною в Інтернеті. З того часу багато чого змінилося. Зараз у доступі багато мов. HTML забезпечує базову структуру веб-сайтів, яка вдосконалюється та модифікується іншими технологіями, такими як CSS та JavaScript. CSS використовується для створення візуалу макета. JavaScript використовується для контролю поведінки різних елементів, та складних анімації. Далі розглянемо кожен з них окремо: – HTML є основою кожної сторінки, незважаючи від складності веб-сайту або

кількості технологій. Ця навичка необхідна для будь-якого веб-розробника. Це точка опори для тих, хто хоче вчитися створювати вміст для Інтернету. HTML розшифровується як HyperText Markup Language. «Мова розмітки» означає, що замість використання мови програмування для виконання функцій HTML використовує теги для ідентифікації різних типів вмісту та цілей, які вони служать для веб-сторінки. Мова розмітки використовує теги, також відомі як «елементи». Теги мають зрозумілі назви: теги заголовків, теги абзаців, теги зображень тощо. Кожна веб-сторінка складається з безлічі цих HTML-тегів, що позначають кожен тип контенту на цій сторінці. Кожен тип контенту на сторінці закритий, оточений HTML-тегами. Якщо тег буде відкрито, то весь вміст, що далі, вважається частиною цього тегу, доки ви не закриєте тег.

При використанні HTML, є можливість додавати заголовки, керувати розривами рядків, складати списки, формувати абзаци, підкреслювати текст, вставляти зображення, створювати посилання, створювати спеціальні символи, керувати деяким стилем, будувати таблиці та багато іншого.

CSS — це каскадні таблиці стилів. Ця мова диктує, які елементи HTML веб-сайту повинні відображатися на сторінці. CSS — це те, що надає стиль вашому веб-сайту.

Тепер поговоримо про JavaScript. JavaScript є складнішою мовою, ніж HTML або CSS, і був випущений у бета-версії до 1995 року. Сьогодні JavaScript підтримує всіх Сучасний веб-браузер, який використовується майже на кожному веб-сайті Отримайте більш потужні та складніші функції. JavaScript — це мова програмування, заснована на логіці, яка може Використовується для зміни вмісту веб-сайту та забезпечення його поведінки Реагувати на дії користувача по-різному. загального користування JavaScript містить прапорці, заклики до дії та додавання нового Ідентифікуйте дані з наявною інформацією. Таким чином, JavaScript є мовою програмування, яка дозволяє веб-розробникам Розробка інтерактивних веб-сайтів. Більшість динамічної поведінки у вас Ви побачите на веб-сторінці завдяки JavaScript, який додає стандартні елементи Керування браузером і поведінка.

1.2. Призначення розробки та галузь застосування

Призначення розробки полягає у створенні повноцінного інформаційного веб-додатку для італійського ресторану. Цей додаток виконує функцію бронювання столиків, дає змогу ознайомитись з меню, контактами та історією закладу.

Цільовою аудиторією є чоловіки та жінки 28-65 років. Аудиторія вищого сегменту

1.3. Підстави до розробки

Відповідно до освітньої програми, згідно навчального плану та графіків навчального процесу, в кінці навчання студент виконує кваліфікаційну роботу. Тема роботи узгоджується з керівником проекту, випускаючою кафедрою, та затверджується з наказом ректора. Таким чином підставами для розробки (виконанням кваліфікаційної роботи) є:

- освітня програма спеціальності 122 “Комп’ютерні науки”;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету “Дніпровська політехніка” № 268-с від 18.05.2022р;
- завдання на кваліфікаційну роботу на тему “Розробка інформаційного веб-додатку для італійського ресторану”.

1.4. Постановка задачі

Мета даного проекту є створення корпоративного веб-додатку.

Веб-додаток повинен відповідати таким умовам:

- Кольори сайту відповідають бренд-буку, а саме червоний градієнт;
- Максимально інтуїтивна карта сайту;

- Можливість користувачу залишати свою інформацію для бронювання столиків;
- Весь дизайн сайту виконати за принципом — панорамність;
- «Респонсивний» адаптив;
- Розширення: 1920px;
- Сітка макету: 12 колон;
- Створення «wireframe»;
- Розробка UserFlow та MindMaps;
- Інтерактивний прототип.

1.5. Вимоги для програми або програмного виробу

1.5.1. Вимоги до функціоналих характеристик

Функціонал веб-додатку полягає у простоті, у тому як користувач може орієнтуватися, як отримати інформацію, яку він шукає, та здобути те, за чим він сюди прийшов. Більшу перевагу віддають тим додаткам, які крім вдалого функціоналу мають привабливий та естетичний дизайн. Але якщо тільки відмінний візуальний дизайн додатку, то він може не досягти реальних результатів. Добрий функціонал та візуал надає значно певні переваги.

1.5.2 Вимоги до інформаційної безпеки

Основні вимоги для інформаційної безпеки :

- забезпечення цілісності даних;
- забезпечення захищеності свого ПЗ від зловмисних програм;
- використання лише безпечного так ліцензованого програмного забезпечення;

- забезпечення конфіденційності інформації;
- забезпечення захищеності програмного продукту від втручання зовні.

Забезпечення інформаційною безпекою корпоративного додатку є одною з найважливіших речей при створенні сайту. Адже від цього залежить конфіденційність даних користувача, та головне за все конфіденційність даних самої компанії. Особливо важливо розуміти, що загроза кіберзлочинності реальна, тож серйозне ставлення до процесу захисту свого веб-додатку є важливим пунктом.

1.5.3. Вимоги до складу та параметрів технічних засобів

Технічними умовами для роботи фінальної версії програмного продукту, а саме веб-додаток для італійського ресторану є:

- обсяг оперативної пам'яті не менш: 500 МБ;
- відеопам'ять : 128 МБ;
- ЦП: Intel-Core Pentium;
- операційна система: Windows, Linux, MacOS, IOS, Android;
- наявність сучасного браузера: Google Chrome, Opera, Safari;
- підтримка високошвидкісного Інтернет з'єднання.

1.5.4. Вимоги до інформаційної та програмної сумісності

Веб-додаток для італійського ресторану відкривається на будь-якій ОС, у будь-якому браузері.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення програми

Функціональне призначення веб-додатку для ресторану «Lucretia», це надання повної інформації про ресторан та його меню, а також можливість залишити свої дані для резервування столиків у самому закладі.

Отже основне призначення:

- перегляд меню;
- ознайомлення користувача з рестораном;
- збір інформації потенційних клієнтів.

2.2 Опис застосованих математичних методів

Під час проектування та розробки веб-додатку для італійського ресторану «Lucretia» не було використано математичні методи, тому що даний сайт їх не потребує.

2.3. Опис використаних технологій та мов програмування

Веб-додаток був створений завдяки наступним технологіям та мов програмування:

- HTML;
- ВЕМ;
- CSS;
- JavaScript.

HTML, або мова гіпертекстової розмітки, дозволяє користувачам Інтернету створювати та структурувати розділи, абзаци та посилання за допомогою елементів, тегів та атрибутів. Однак варто зазначити, що HTML не вважається мовою програмування, оскільки він не може створювати динамічні функції.

HTML має багато варіантів використання, а саме:

- Веб-розробка. Розробники використовують HTML-код, щоб розробити, як браузер відображає елементи веб-сторінки, такі як текст, гіперпосилання та мультимедійні файли.
- Інтернет навігація. Користувачі можуть легко переміщатися та вставляти посилання між пов'язаними сторінками та веб-сайтами, оскільки HTML активно використовується для вбудовування гіперпосилань.
- Веб-документація. HTML дозволяє впорядковувати та форматовувати документи, подібно до Microsoft Word.

Також варто зазначити, що HTML тепер вважається офіційним веб-стандартом.

BEM — це інтерфейсний метод іменування для організації та іменування класів CSS. Методологія блоку, елемента, модифікатора — це популярна конвенція щодо найменування імен класів у HTML і CSS. Це допомагає писати чистий CSS, дотримуючись простих правил.

Проекти будь-якого розміру з CSS можуть отримати вигоду від BEM-фреймворку, якщо не писати стилі безпосередньо у вже організовані файли JavaScript і не використовувати стилізовані компоненти або щось подібне. Щоб доповнити BEM, додайте деякі частини SMACSS, які діють як посібник зі стилю для умовних імен. BEM легко читається і впорядковує стилі, навіть якщо проекти надзвичайно великі.

Іменування класів ніколи не було легкою частиною CSS. Зовнішнім розробникам важко зрозуміти, що роблять визначені розробником класи. Умови про іменування можуть швидко вийти з-під контролю під час рефакторингу коду кілька разів і під час роботи з командою.

Більшість проблем виникають через погане іменування та відсутність області дії (так само як область, яка закриває контекст). Наприклад, при читанні коду меню розробники повинні бути впевнені, що код не містить жодних стилів заголовків. У цій статті я розповім, що таке BEM, як він працює, і рекомендації щодо успішного впровадження BEM.

Каскадні таблиці стилів, які ще називають CSS, — це проста мова дизайну, призначена для спрощення процесу створення презентаційних веб-сторінок.

CSS обробляє зовнішній вигляд і відчуття частини веб-сторінки. Використовуючи CSS, можна керувати кольором тексту, стилем шрифтів, інтервалом між абзацами, розміром і розташуванням стовпців, фоновими зображеннями або кольорами, дизайном макета, варіантами відображення для різних пристроїв і розмірів екрана. а також безліч інших ефектів.

JavaScript — це мова програмування, яку використовують переважно веб-браузери для створення динамічного та інтерактивного інтерфейсу для користувача. Більшість функцій і додатків, які роблять інтернет незамінним у сучасному житті, закодовані в тій чи іншій формі JavaScript.

Найперші втілення JavaScript були розроблені наприкінці 1990-х для веб-браузера Netscape Navigator. У той час веб-сторінки були статичними, пропонуючи невелику взаємодію з користувачем, крім натискання посилань і завантаження нових сторінок. Вперше JavaScript увімкнув анімацію, адаптивний вміст і перевірку форми на сторінці.

Також JavaScript, це те, що відомо як клієнтський скрипт. Більшість веб-додатків, таких як пошукова система, працюють за допомогою взаємодії між пристроєм користувача (наприклад, комп'ютером, телефоном або планшетом) і віддаленим сервером. Програмне забезпечення на віддаленому сервері надсилає інформацію клієнту (тобто на машину користувача), а програмне забезпечення на стороні клієнта зчитує інформацію та відображає веб-сторінку на екрані.

Сценарій на стороні клієнта — це мова програмування, яка повністю виконує свої завдання на машині клієнта і не вимагає взаємодії з сервером для функціонування. Наприклад, якщо на вашому комп'ютері завантажено веб-сторінку, а ваш провайдер не працює, ви все одно можете взаємодіяти з веб-сторінками, які вже завантажені у вашому браузері. Однак ви не зможете отримати доступ до нових веб-сторінок або отримати доступ до даних, розташованих віддалено.

Деякі з динамічних покращень веб-сайту, які виконує JavaScript:

- Автозавершення
- Завантажуйте новий вміст або дані на сторінку без перезавантаження сторінки
- Ефекти прокрутки та спадні меню
- Анімуйте елементи сторінки, такі як вицвітання, зміна розміру або переміщення
- Відтворення аудіо та відео
- Перевірка введених даних з веб-форм
- виправити проблеми з сумісністю браузера

Хоча JavaScript є клієнтською мовою, деякі з його найпотужніших функцій передбачають асинхронну взаємодію з віддаленим сервером. Асинхронний просто означає, що JavaScript може спілкуватися із сервером у фоновому режимі, не перериваючи взаємодії користувача на передньому плані.

Візьмемо, наприклад, пошукову систему. Сьогодні майже всі пошукові системи мають функцію автозаповнення. Користувач починає вводити слово у вікно пошуку, а нижче з'являється список можливих пошукових термінів або фраз. Досвід безперешкодний. Запропоновані пошукові терміни відображаються без перезавантаження сторінки.

У фоновому режимі JavaScript читає електронні листи, коли користувач їх вводить, надсилає ці листи на віддалений сервер, а сервер надсилає пропозиції назад.

Програмне забезпечення на стороні сервера аналізує слова та запускає алгоритми, щоб передбачити пошуковий термін користувача. Такі програми диявольсько великі і складні. JavaScript на клієнтській машині максимально простий і невеликий, щоб не сповільнювати взаємодію з користувачем. Зв'язок між JavaScript і програмою на стороні сервера обмежена пропускнуою здатністю користувача. Ось чому розробники віддають перевагу ефективності функцій JavaScript і зберігають якомога менший обсяг даних, що передаються між програмами.

Тільки після того, як користувач вибере пошуковий термін, вся сторінка перезавантажується і повертає результати пошуку. Такі механізми, як Google, зменшили або виключили необхідність перезавантаження навіть для цього кроку. Вони просто видають результати, використовуючи той самий асинхронний процес.

2.4. Опис структури системи та алгоритмів її функціонування

Інформаційний веб-додаток для італійського ресторану «Lucretia», складається з двох частин:

1. UX/UI design;
2. Front-end — development;

Перша частина складається з фаз дослідження взаємодій (рис.2.1.)



Рис. 2.1. Фази дослідження взаємодій

Аналіз — сформування загального уявлення про продукт, його аудиторію, задачі та обмеження.

Етапи аналізу:

- Польові дослідження;
- Інтерв'ю з користувачами;
- Інтерв'ю з учасниками процесу;
- Метод ведення щоденника;
- Збір вимог і обмежень від клієнта.

Проектування — визначення конкретних механік для досягнення цілей.

Етапи проектування:

- Вивчення продукту конкурентів;

- Розробка персон;
- Аналіз шляху клієнту;
- Розробка сценаріїв використання (рис.2.2.);
- Дослідження за допомогою сортування карток;
- Тестування прототипу.

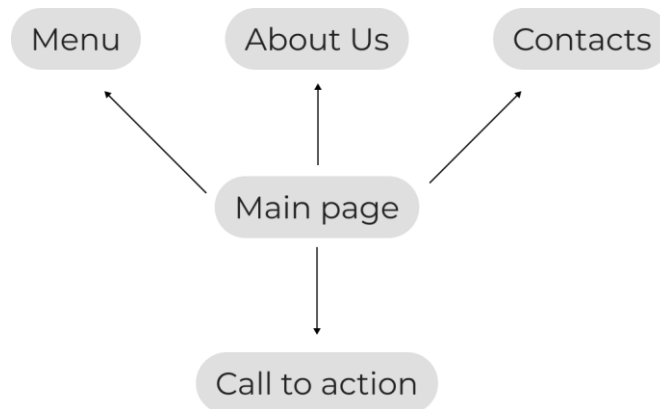


Рис. 2.2. Сценарій використання

Дизайн — аналіз зручності та зрозумілості концепції.

Етапи дизайну:

- Мудбоард;
- Дослідження останніх UI трендів;
- Розробка концепції;
- Розробка дизайн-системи.

Друга частина складається з розробкою програмного коду. Починаємо зі створення папки у відповідному диску (рис. 2.3.)

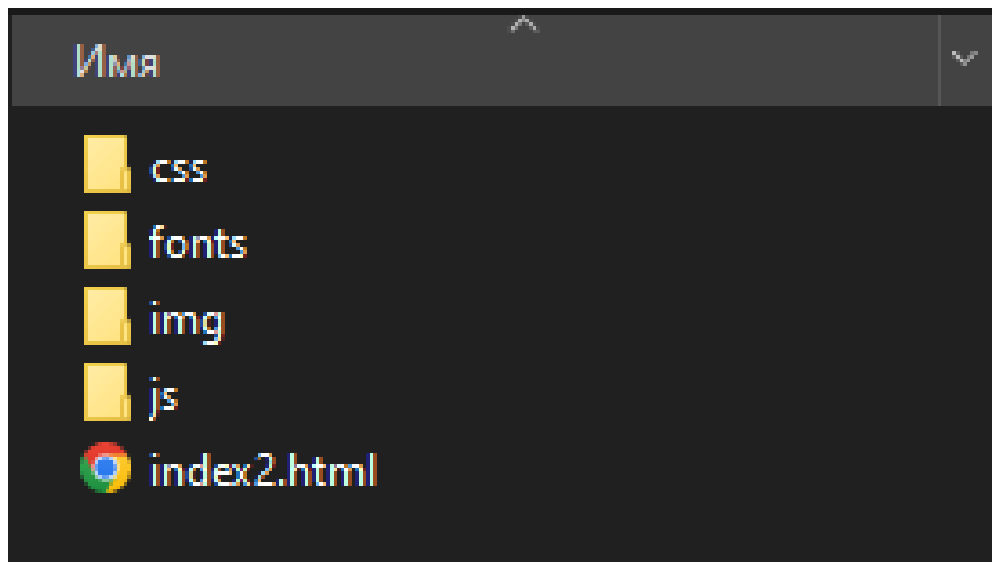


Рис. 2.3 папка з елементами для зручної розробки

Далі перекидаємо цю папку до свого IDE редактору (рис. 2.4.)

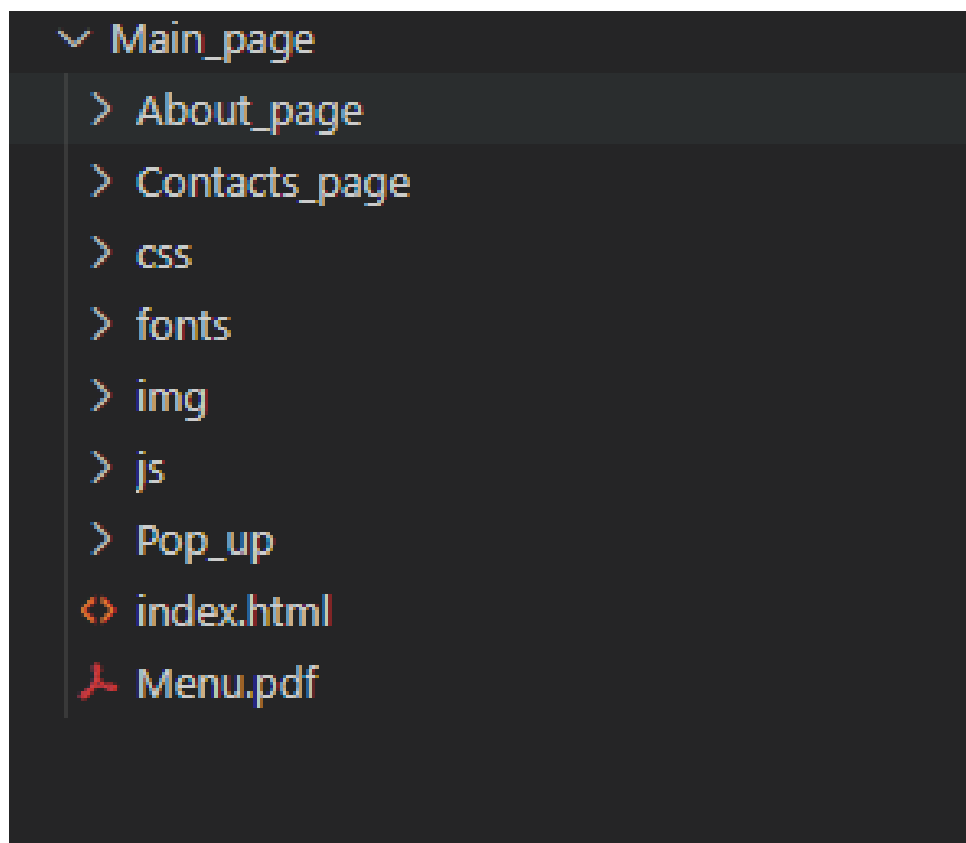


Рис. 2.4. Структура веб-додатку

Папка `Main_page` з якої запускається основна частина додатку, вона містить у собі адресу HTML, каскадні стилі CSS, папку з фото елементами, скрипти JavaScript який підключений у «`index.html`» через тег (`<script src="">` `</script>`).

У папці css (рис.2.5.), знаходиться файл, який описує дизайн веб-додатку

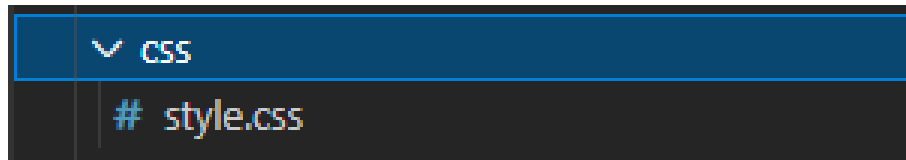


Рис. 2.5. Структура папки css

Для збереження фото елементів була створена папка img (Рис. 2.6.), вона містить у собі не тільки фото елемента, а й ще векторні зображення (наприклад логотип ресторану).

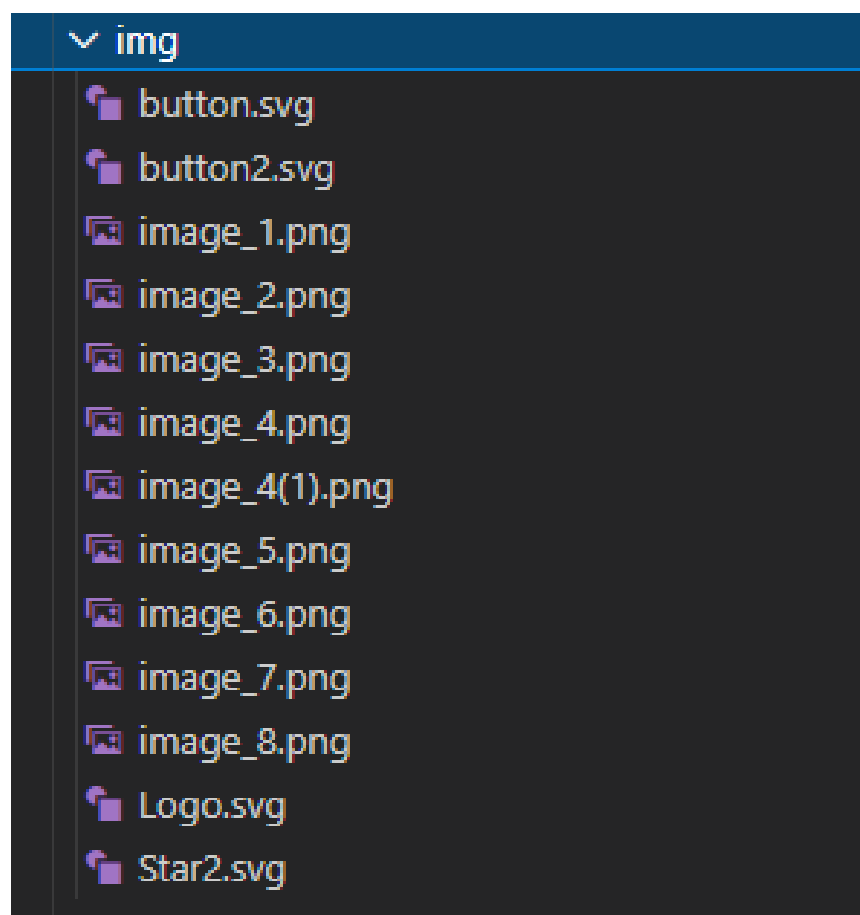


Рис. 2.6. Структура папки img

Також є папка js (рис. 2.7.), яка містить у собі скрипт, який дає змогу зробити плавний слайдер для фото галереї, а ще там знаходиться скрипт, який викликає та закриває pop up для бронювання .

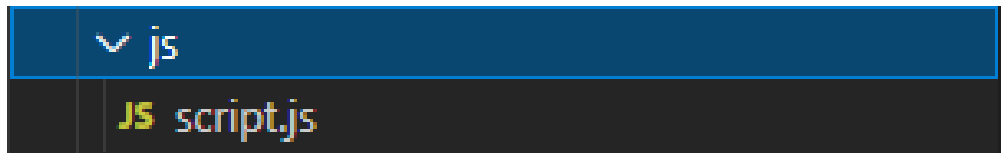


Рис. 2.7. Структура папки js

Файл Menu.pdf (рис 2.8.), містить у собі інформацію стосовно меню ресторану, він відкривається окремим вікном у браузері, як звичайний pdf формат.

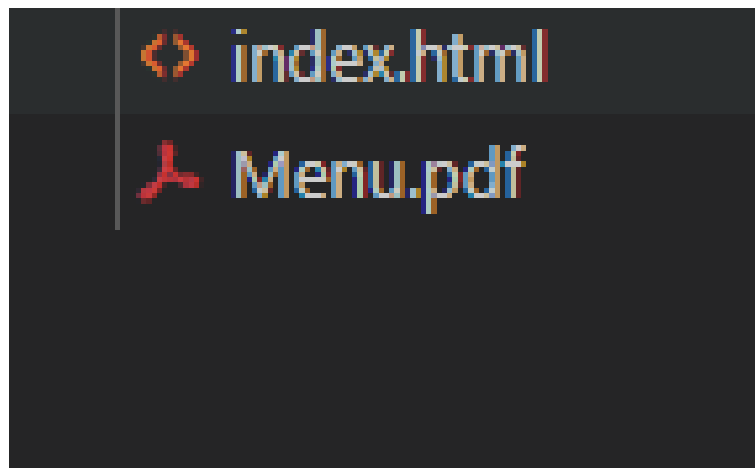


Рис. 2.8. pdf-файл

Папка About_page (рис. 2.9.), це окрема сторінка про історію виникнення ресторану. За структурою вона така ж як Main_page.

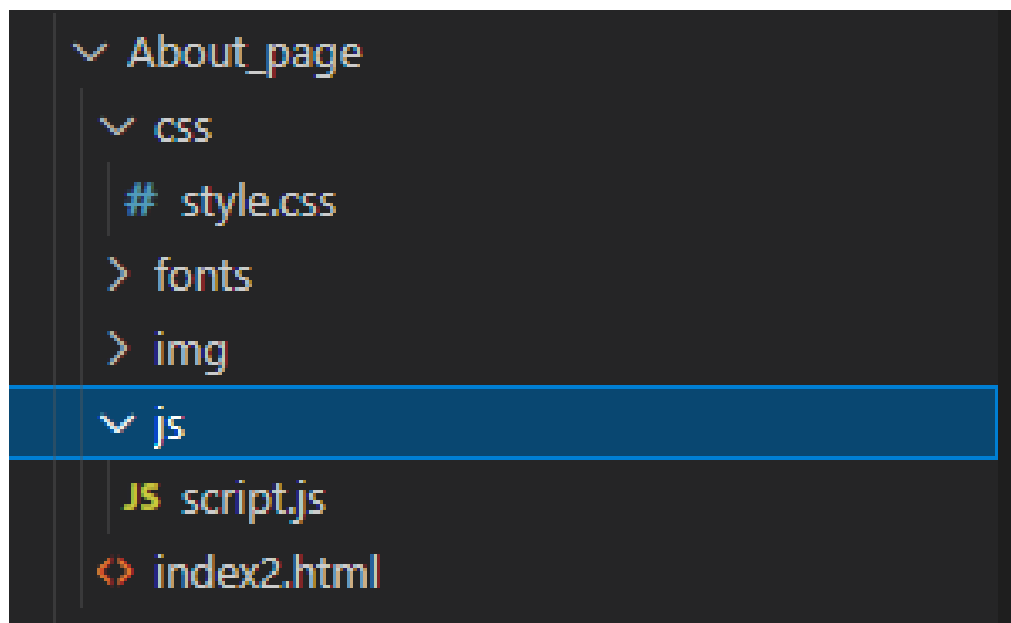


Рис. 2.9. Структура папки About_page

Наступна папка `Contacts_page` (рис. 2.10.), окрема сторінка з інформацією про адресу ресторану тощо.

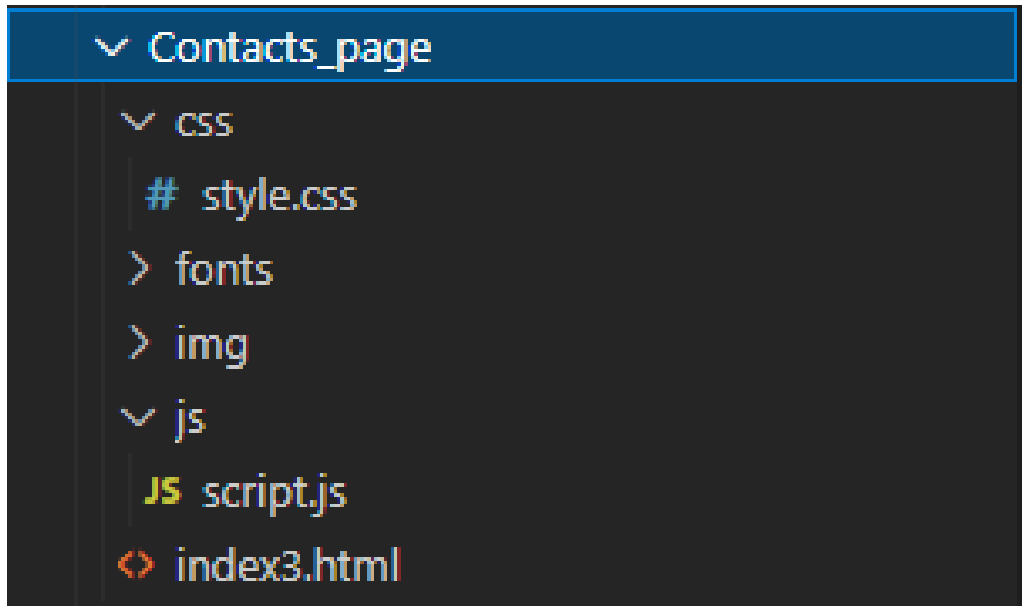


Рис. 2.10. Структура папки `Contacts_page`

2.5. Обґрунтування та організація вхідних та вихідних даних програми

Є тільки єдиний вхід даних у кваліфікаційній роботі, це коли користувач хоче забронювати столик у ресторані, та йому потрібно вести дані свого ім'я та номер телефону (рис. 2.11.)

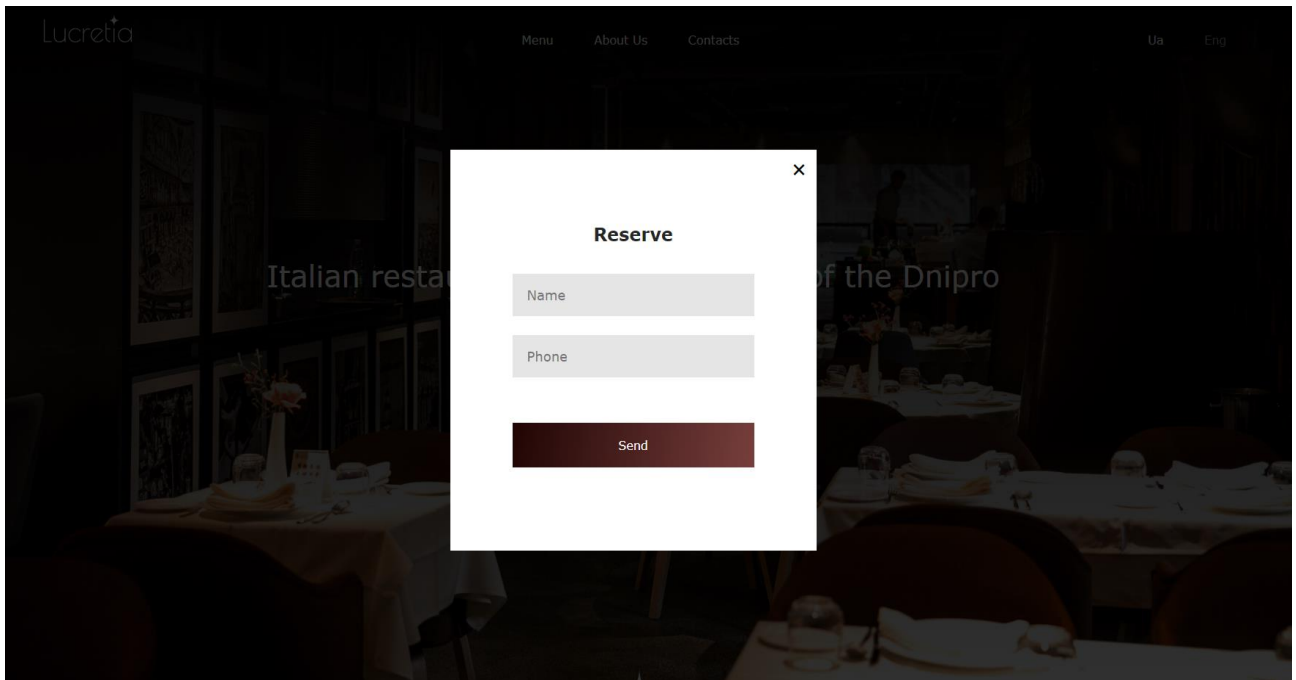


Рис. 2.11. Приклад вхідної інформації імені та номеру телефону

Також вхідними даними в інформаційному веб-додатку є файли cookies. Це текстовий файл, який містить, певний час, у собі інформацію вашого браузера. Такі файли бувають сеансові та постійні.

2.6. Опис розробленої системи

2.6.1. Використані технічні засоби

Для створення та тестування інформаційної системи була використана персональна ЕОМ, з наступними системними характеристиками (рис. 2.12.):

- Процесор: Intel-Core i5 9400F 2.90GHz;
- Оперативна пам'ять: 8Гб
- Відеокарта: Geforce GTX 1660 ti 6Гб;
- Операційна система: Windows 10 Pro;
- 1Гб вільного місця на Ssd диску;
- Клавіатура та комп'ютерна миша;
- Монітор.

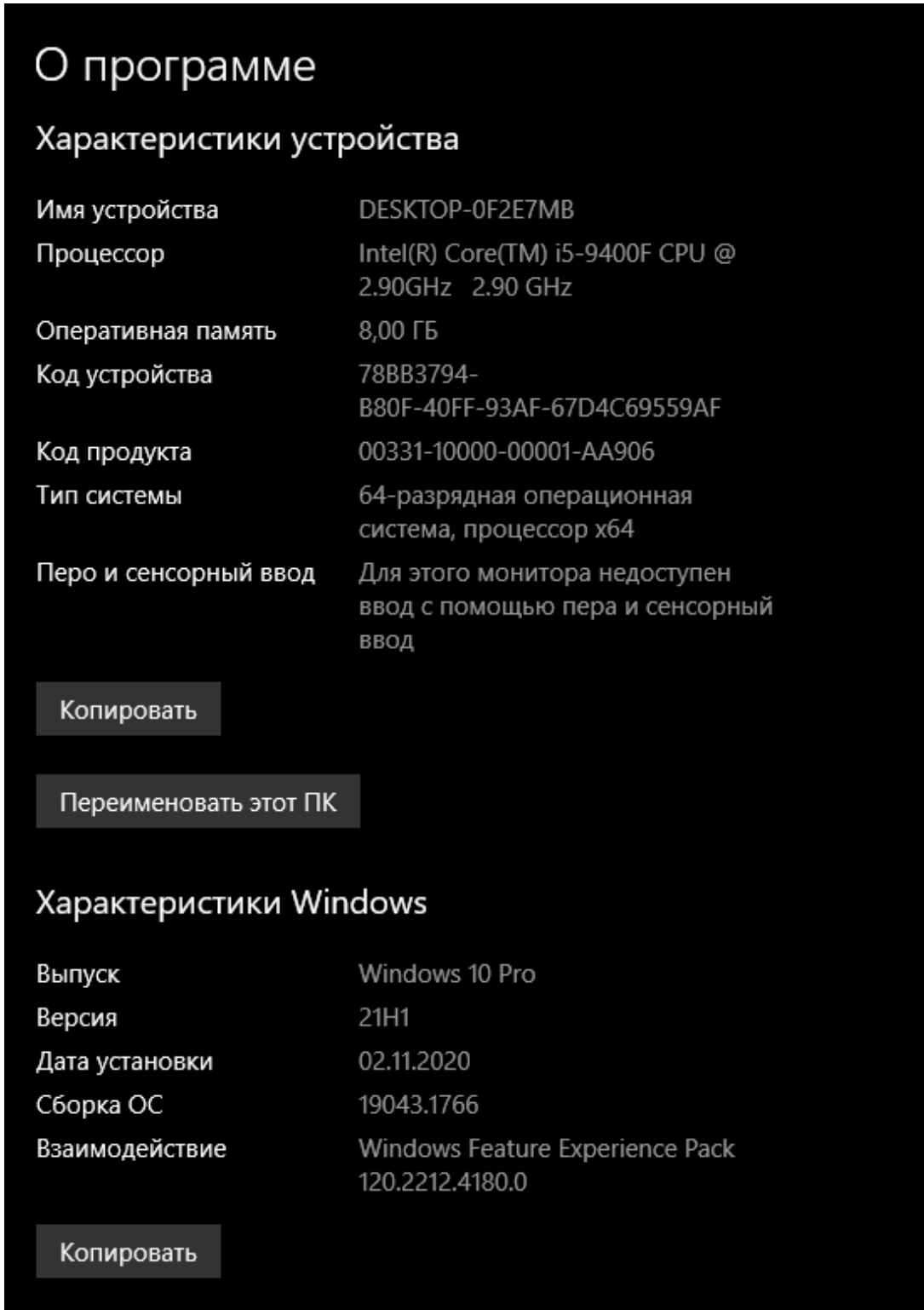


Рис. 2.12. Системні характеристики

2.6.2. Використані програмні засоби

Редактор вихідного коду для створення інформаційного веб-додатку був обраний — Visual Studio Code. (рис. 2.13.)

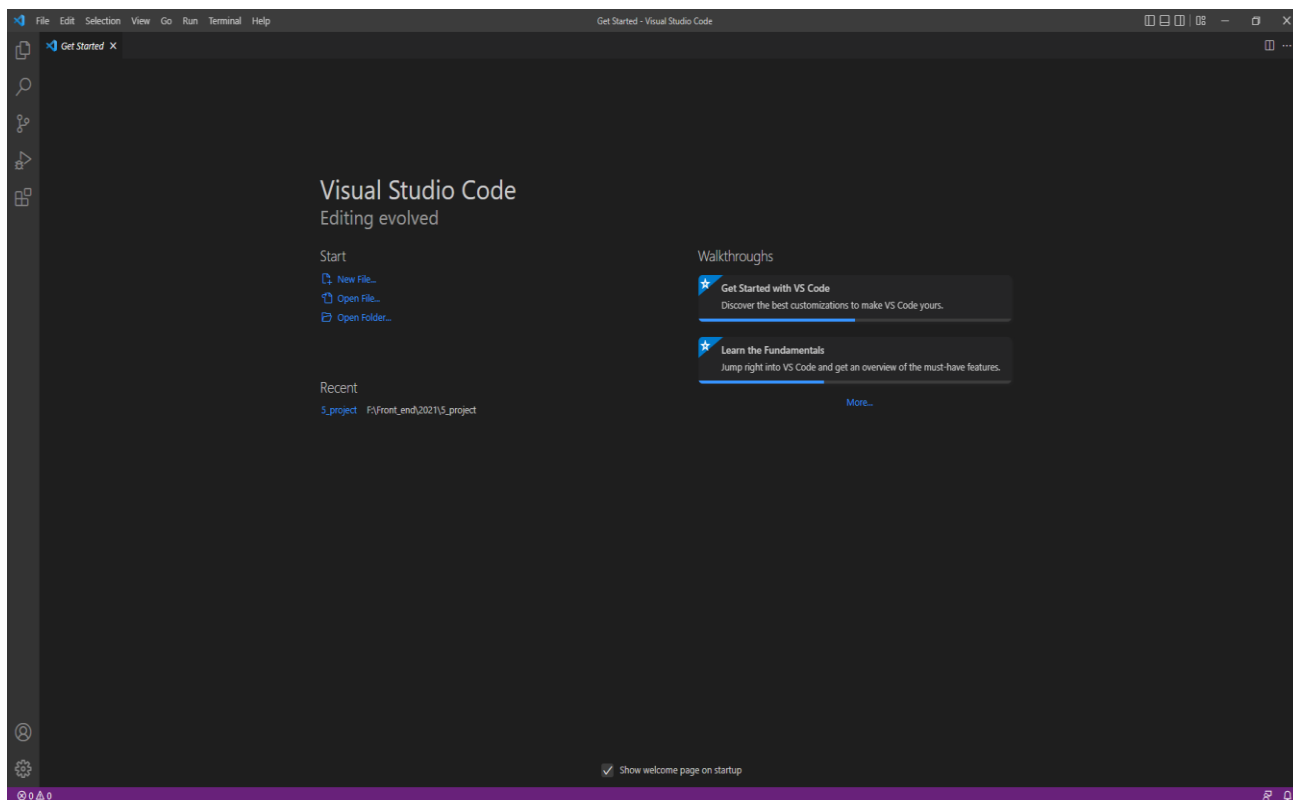


Рис. 2.13. Visual Studio Code

Visual Studio Code або VS Code — це безкоштовний текстовий редактор з відкритим вихідним кодом від Microsoft. Цей редактор доступний для усіх операційних систем, таких як Windows, Linux і macOS. Редактор є відносно легким, але він містить деякі потужні функції, які зробили VS Code одним із найпопулярніших інструментів середовища розробки останнім часом.

VS Code підтримує багато мов програмування, таких як Java, C++, Python, CSS, Go і Dockerfile, тощо. Крім того, VS Code дає змогу додавати і створювати нові розширення, включаючи літери коду, налагоджувачі, а також підтримку хмарної та веб-розробки.

Також для комфортної роботи, я використовував спеціальні розширення для платформи VSCode — це Live Server (рис.2.15.). Він дає змогу бачити зміни наживо.

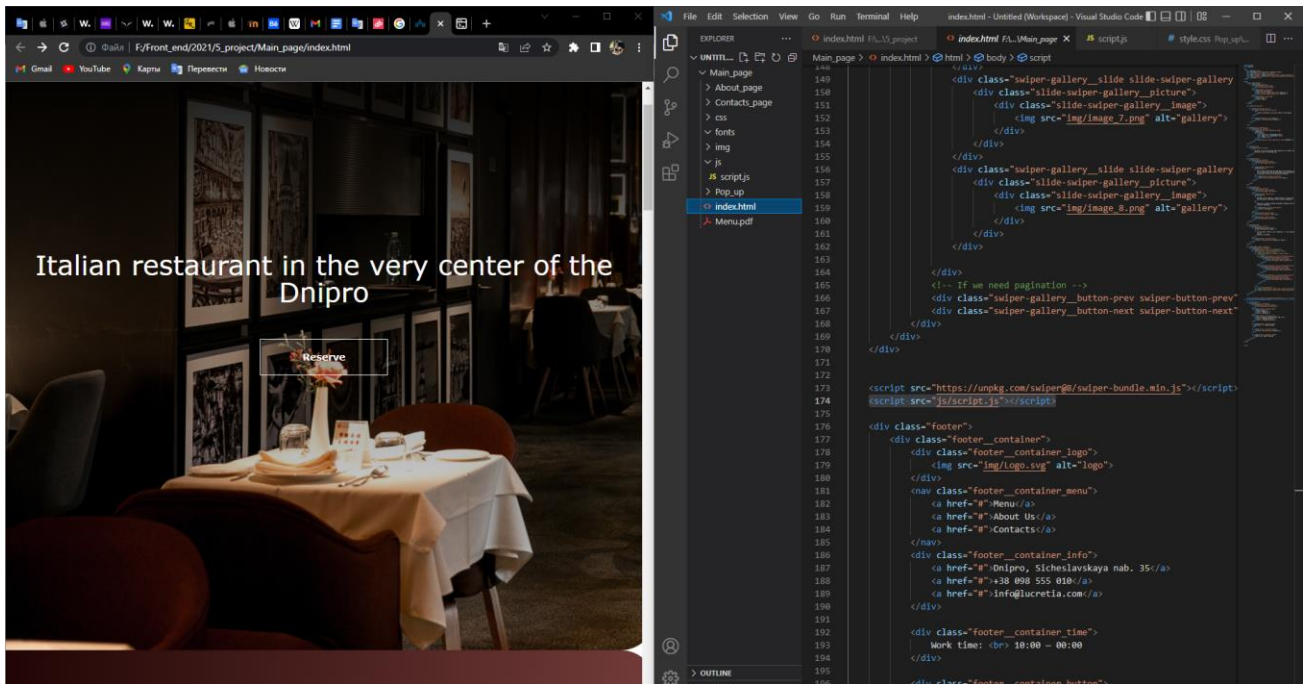


Рис. 2.15. Розширення Live Server

При створенні макету використовується багато редакторів, а саме Figma (рис. 2.16.) є лідером серед інших.

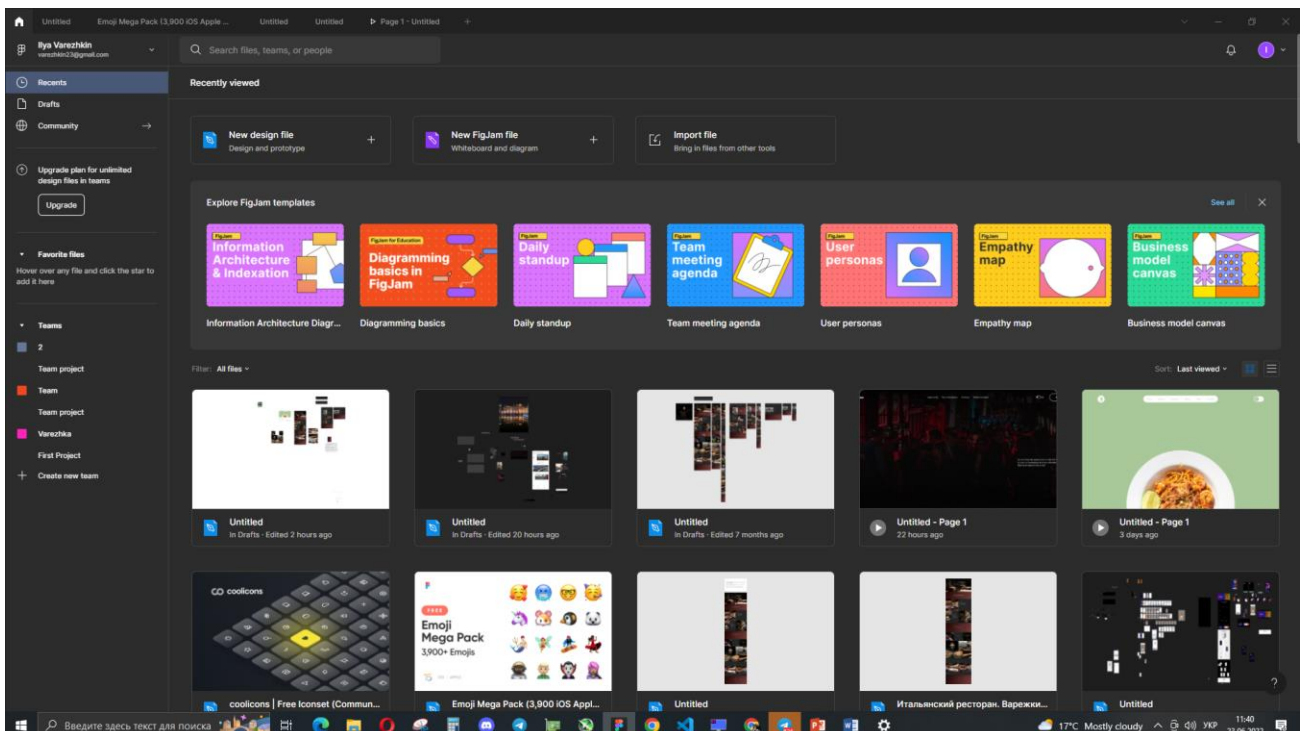


Рис. 2.16. Головна сторінка застосунку Figma

Figma — векторний сервіс для розробки та прототипування інтерфейсів.

Мабуть зараз цей редактор самий популярний серед усіх редакторів, тому що він дуже зручний та зрозумілий, у ньому присутні тільки самі важливі інструменти, для чіткого створення будь-якого інтерфейсу. Цей редактор працює через браузер, або можна скачати додаток до вашої операційної системи.

Ще розробці я використовував корисний інструмент, який є у браузері Chrome. Цей інструмент називається Chrome Web Tools (рис. 2.17.), та дає змогу бачити весь код front-end. Він також безкоштовний.

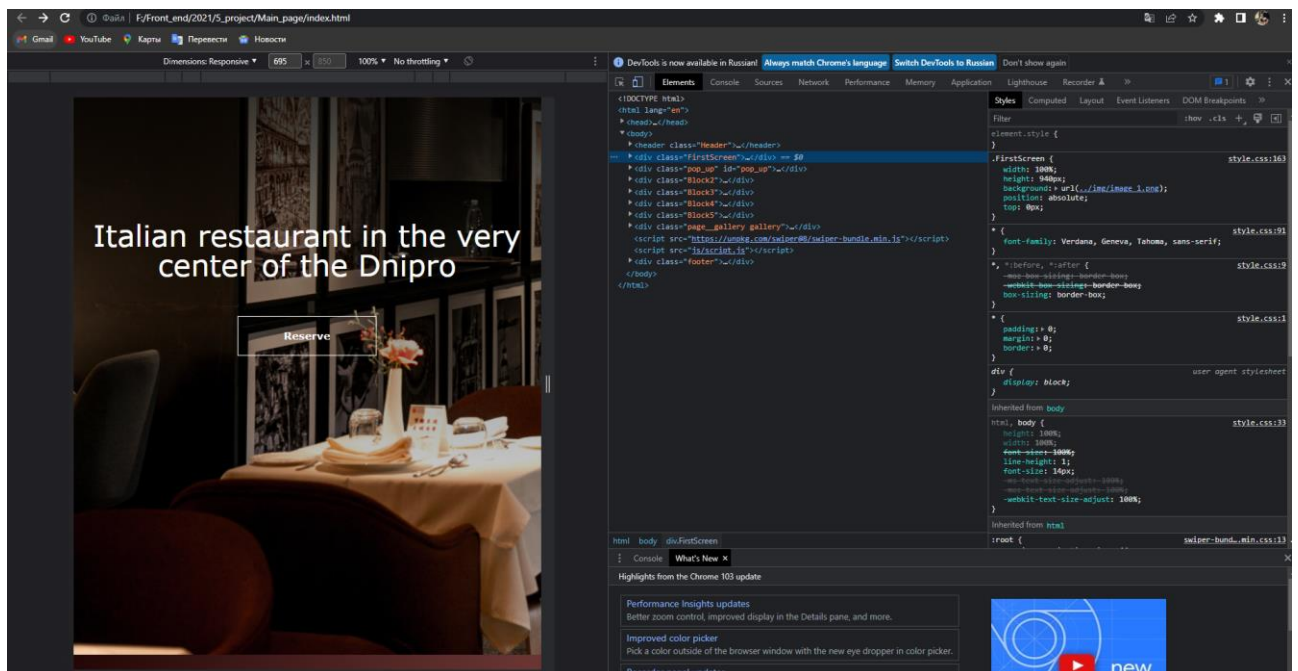


Рис. 2.17. Інструмент Chrome Web Tools

2.6.3. Виклик та завантаження програми

Для використання сайту, потрібно відкрити файл index.html через браузер. А щоб повноцінно користуватись сайтом, потрібно розробити серверну частину, тобто (backend). Потім завантажити на хостинг та придбати доменне ім'я.

2.6.4. Опис інтерфейсу користувача

Після запуску з'являється головна сторінка веб-додатку італійського ресторану «Lucretia». Одразу стає видно логотип, хедер та перший екран с закликком на бронювання столику. (рис 2.18)

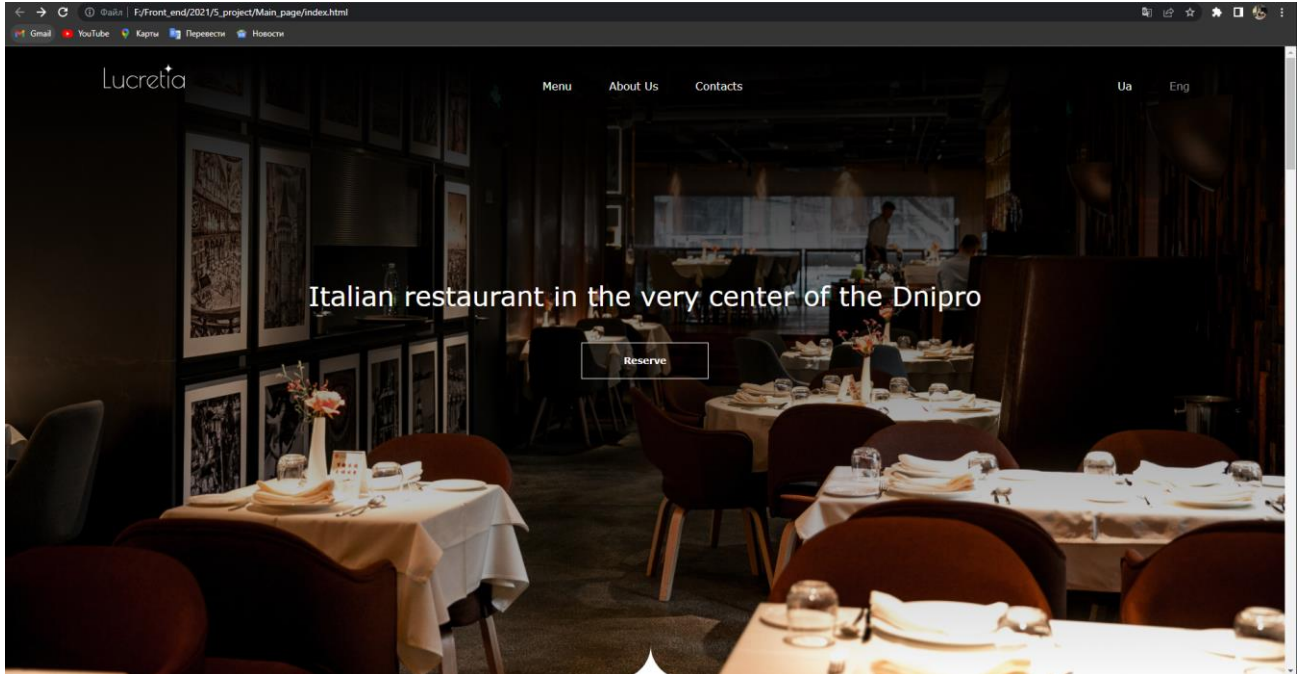


Рис. 2.18. Перший екран веб-додатку

При натисканні на кнопку «Reserve» з'являється модальне віконце (рис. 2.19.) з вводом інформації потрібної для бронювання столиків. Після цього, інформація поступає до адміністратора сайту і потім адміністратор передає інформацію до call-центру. З call-центру передзвонюють користувачу, котрий залишав свої данні, та вони домовляються за бронювання.

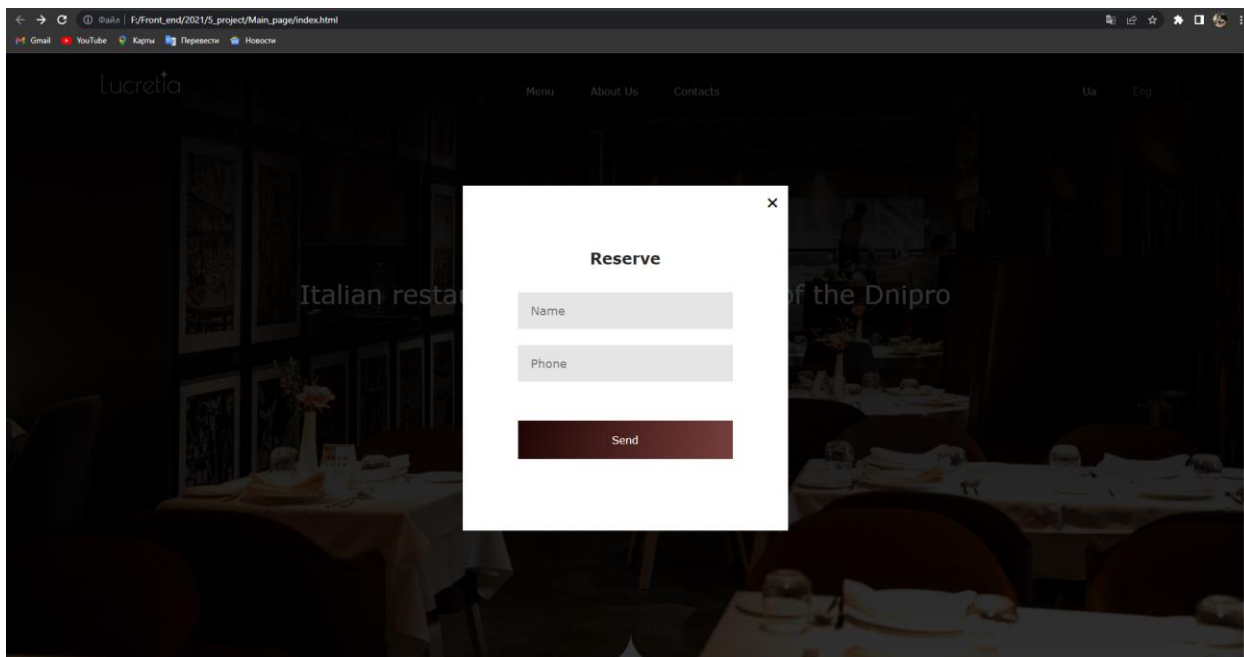


Рис. 2.19. Форма для бронювання

Після першого екрану іде блок з поясненням про ресторан (рис. 2.20.), що в ньому присутня душа, що це не просто ресторан, а це цікава історія. Це є такий концепт ресторану — теплі спогади, та тепла атмосфера.

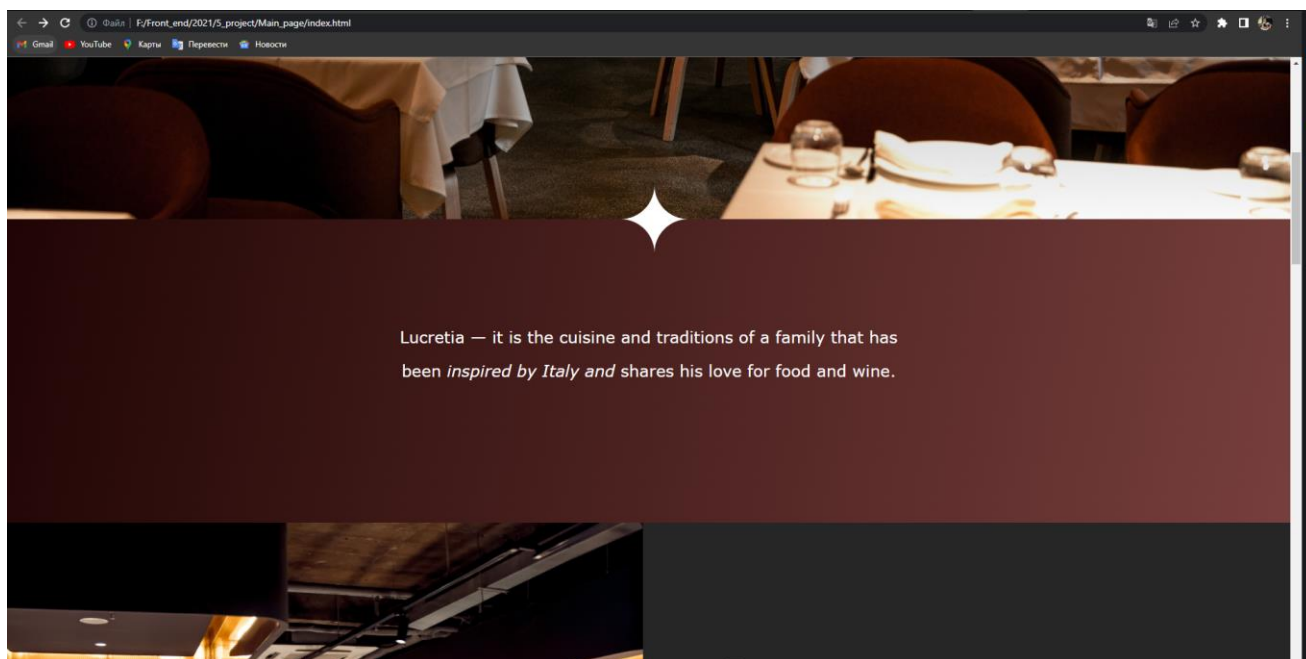


Рис. 2.20. Наступний блок після першого екрану.

Далі іде блок, про більш розгорнуту розповідь про ресторан (рис. 2.21.), тут також присутня кнопка для переходу на іншу сторінку, де максимально детально пояснюють про ресторан.

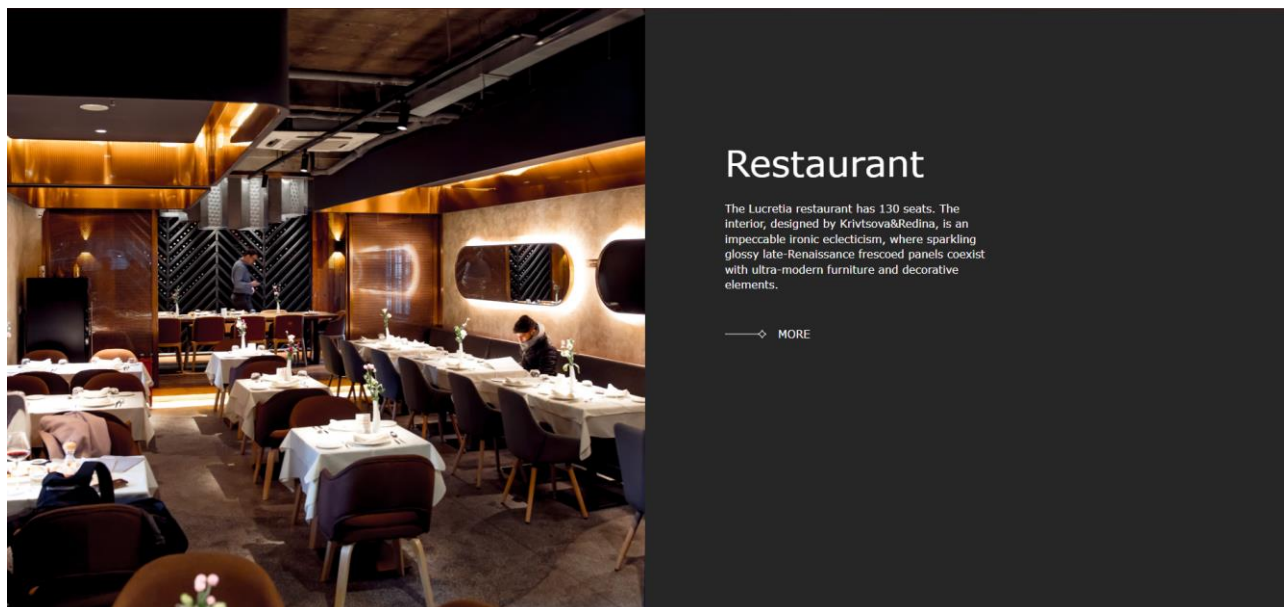


Рис. 2.21. Блок про ресторан

Скролимо нижче, та бачимо блок про меню (рис. 2.22.), тут описується про концепт меню та кнопка з переходом до іншої сторінки з повним описанням меню.

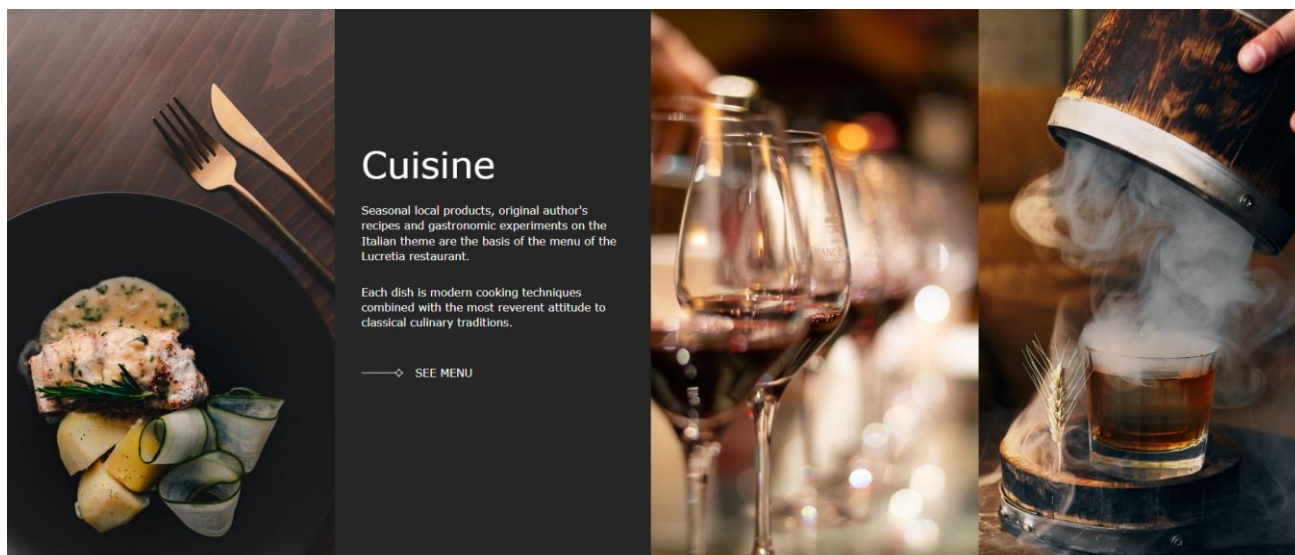


Рис. 2.22. Опис концепції меню

Ще нижче скролимо, і далі іде блок з викликом дії (call to action, рис.2.23.) Цей блок для того, щоб користувач не втрачав уваги до резервування, бо це є пріоритетом номер один.

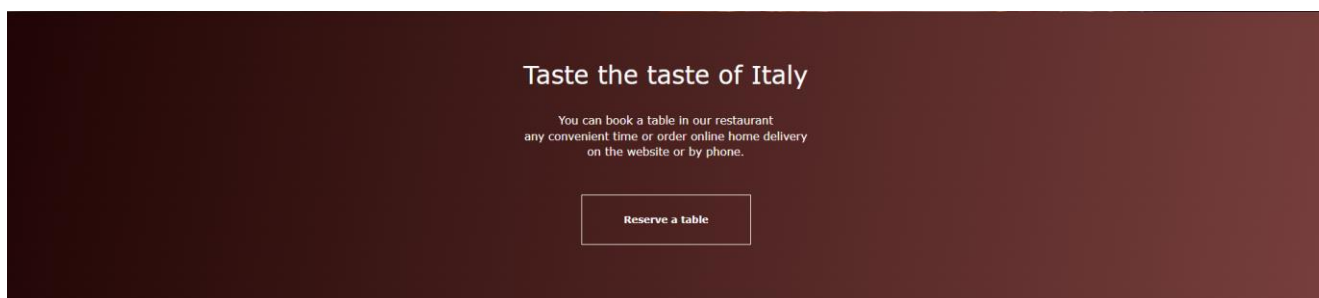


Рис. 2.23. Блок «Call to action»

Потім, передостанній блок — галерея (рис. 2.24.) тут з допомогою javascript створено плавний слайдер.

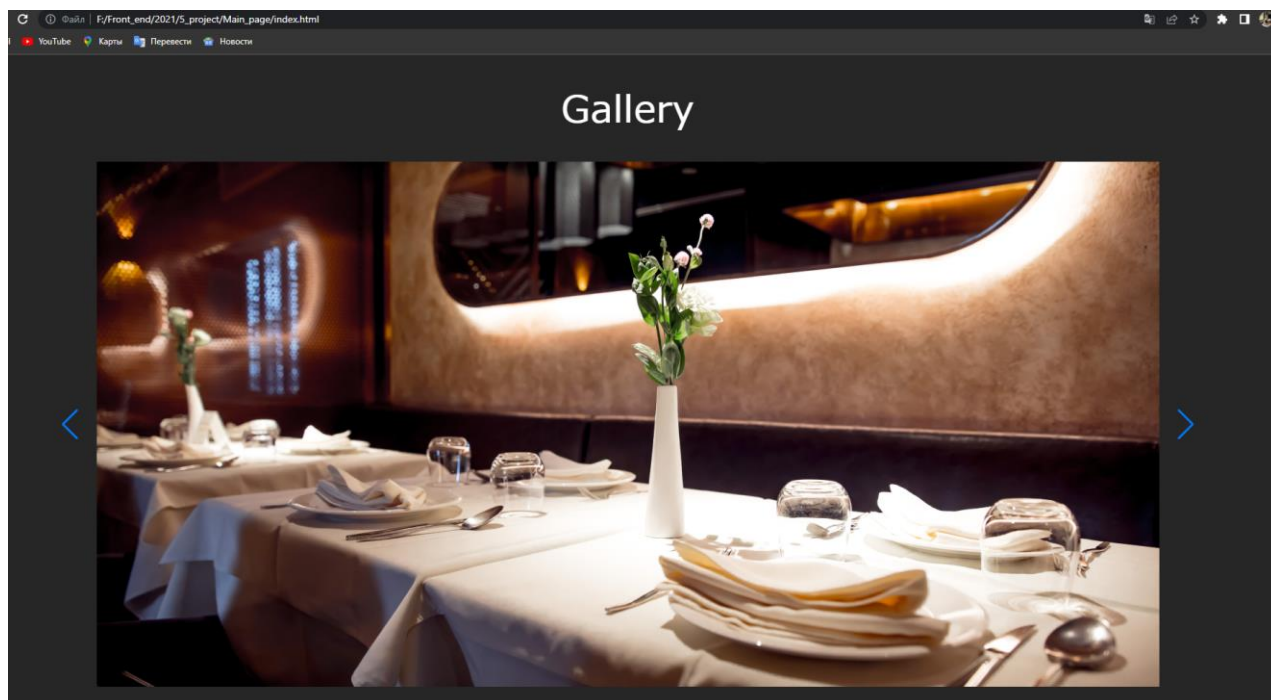


Рис. 2.24. Галерея

І нарешті останній блок, це футер (рис. 2.25.), який є один з обов'язкових елементів сайту. Тут присутня уся потрібна інформація для користувача.

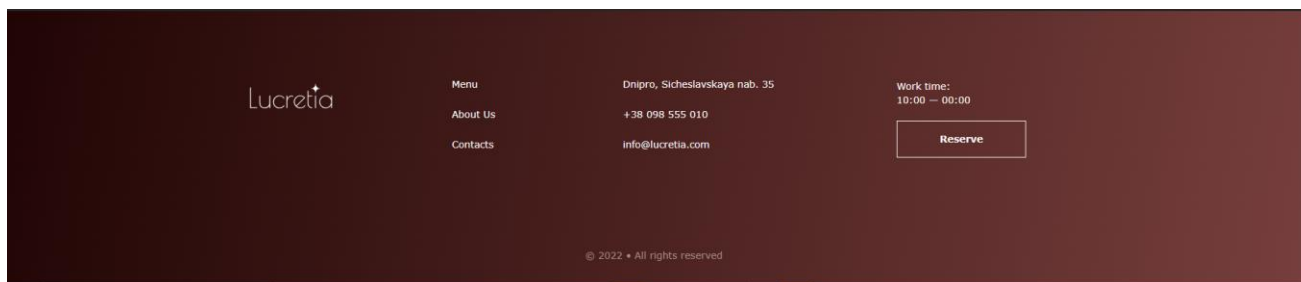


Рис. 2.25. Футер

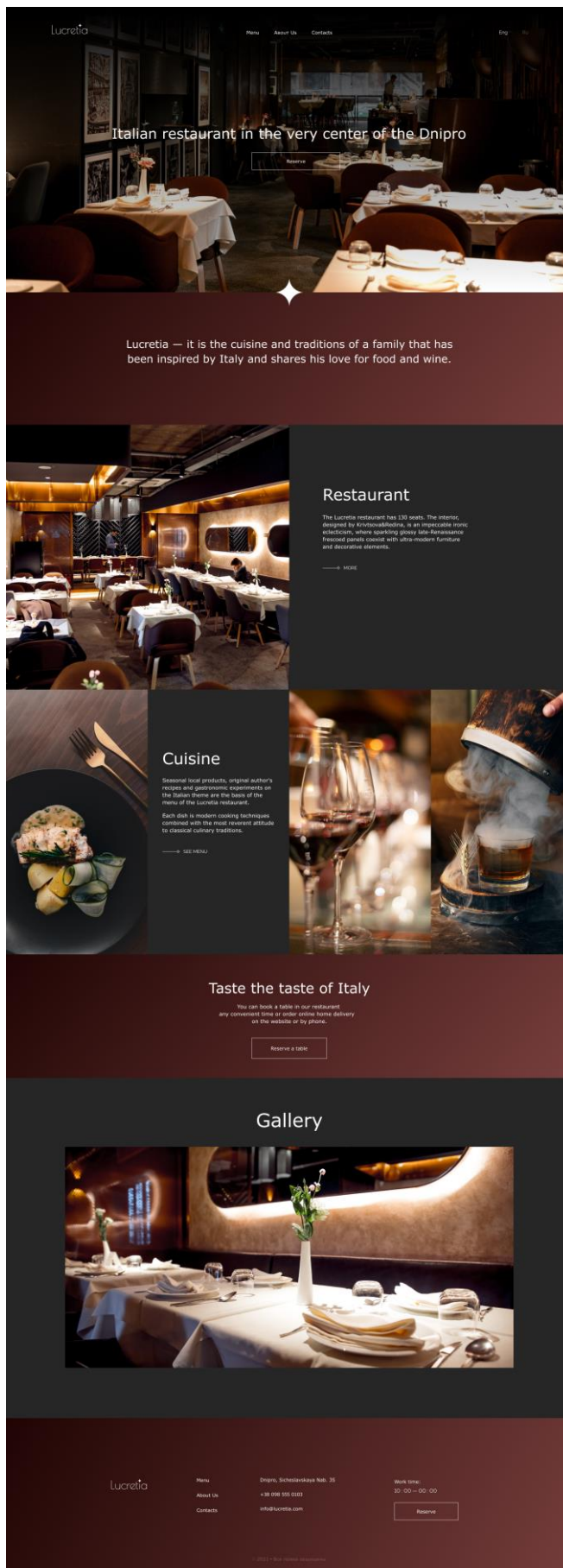


Рис. 2.26. Головна сторінка увесь зріст

About us



Gino De Luca

founder of Lucretia

«Italian cuisine is limitless for me, as it is very unique in different regions, from the Italian Alps to the island of Sicily. Representing Italian cuisine in Ukraine, for me, as for every Italian, it is important to put love into creating a gastronomic masterpiece so that the guest receives a freshly prepared dish in a modern and appetizing serving of quality products.»

Foundation

Lucretia — legendary Italian restaurant, founded in 2003, with an interesting long history, high-class service, exquisite Italian cuisine with Mediterranean motives, fresh seafood and the best meat in the world.



Interior

The Lucretia restaurant has 130 seats. The interior was designed by Krivtsova&Redina and is an impeccable ironic eclecticism, where sparkling glossy panels with late renaissance frescoes coexist with ultra-modern furniture and decorative elements.

Рис. 2.27. Страница «About Us»

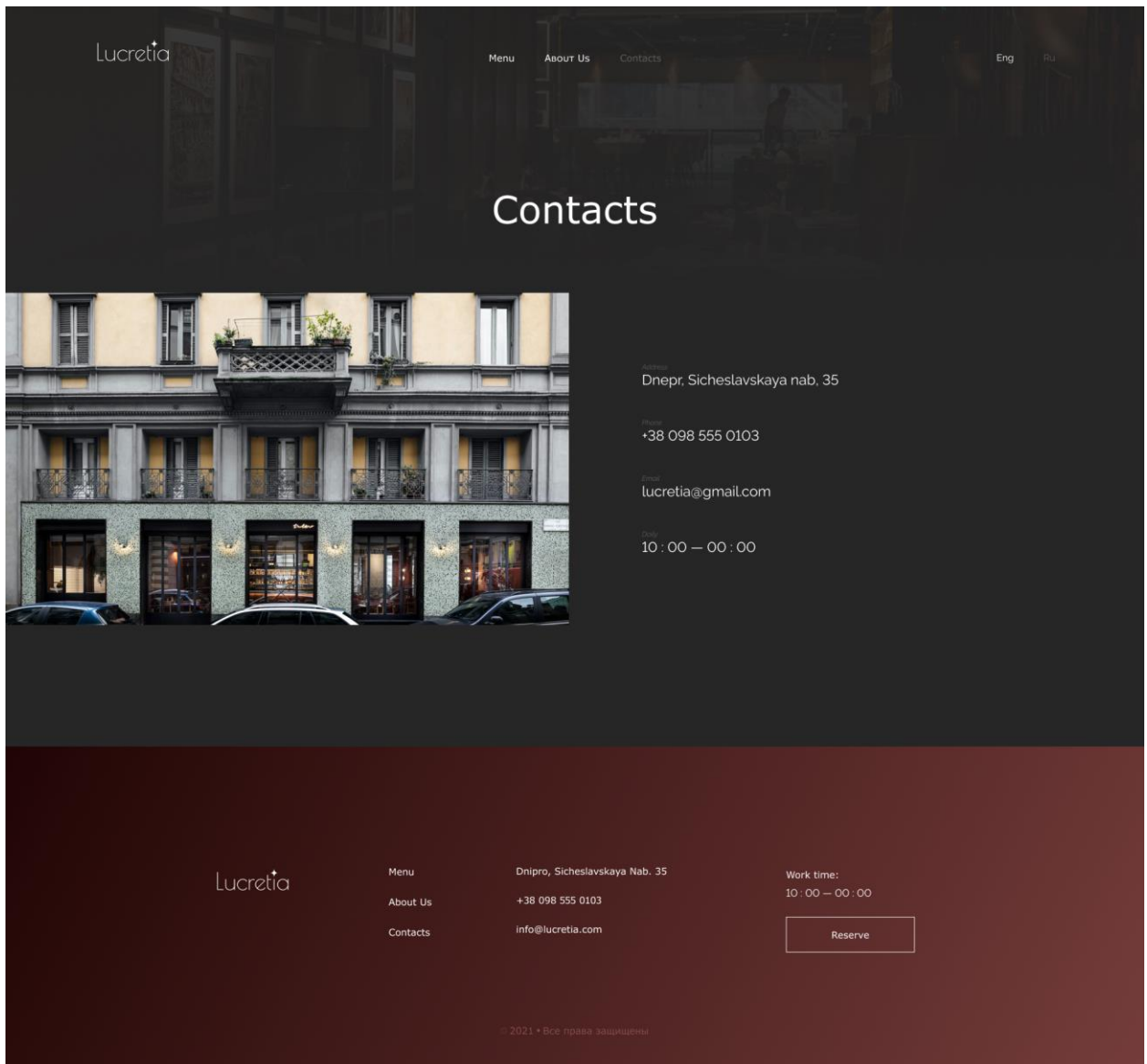


Рис. 2.28. Сторінка «Contacts»

РОЗДІЛ 3

ЕКОНОМІЧНИЙ РОЗДІЛ

3.1 Розрахунок трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. передбачуване число операторів програми – 1630;
2. коефіцієнт складності програми – 1,1;
3. коефіцієнт корекції програми в ході її розробки – 0,08;
4. годинна заробітна плата програміста – 113грн/год;

Середня зарплата програміста на позиції junior front-end development становить 20 000грн на місяць по даним сайту Work.ua. Отже, 20 000грн / 22 (робочих днів у місяць) = приблизно 910грн, тобто середня зарплата за годину становить приблизно 113грн.

5. коефіцієнт збільшення витрат праці в наслідок недостатнього опису задачі – 1,2;

6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1;

7. вартість машино-годин ЕОМ – 20 грн/год.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{oml} + t_d, \text{ людино-годин,}$$

(3.1)

де: t_o — витрати праці на підготовку й опис поставленої задачі;

t_a — витрати праці на дослідження алгоритму рішення задачі;

t_n — витрати праці на розробку блок-схеми алгоритму;

t_{oml} — витрати праці на налагодження програми на ЕОМ;

t_d — витрати праці на підготовку документації

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \cdot C \cdot (1 + p), \quad (3.2)$$

де: q — передбачуване число операторів (1630);

C — коефіцієнт складності програми (1.1);

p — коефіцієнт кореляції програми в ході її розробки (0.08).

Підставивши відомі дані в формулу маємо:

$$Q = 1.1 \cdot 1630 \cdot (1 + 0.08) = 1936.44$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75..85) \cdot k}, \text{ людино-годин,} \quad (3.3)$$

де: B - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

k - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи зданої спеціальності.

$$t_u = (1936.44 \cdot 1.1) / (75 \cdot 1) = 28.4 \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = \frac{Q}{(20...25) \cdot k}, \text{ людино-годин} \quad (3.4)$$

де: Q — умовне число операторів програми;

k — коефіцієнт кваліфікації програміста.

Підставивши дані значення в формулу (3.4), отримаємо:

$$t_a = 1936.44 / (20 \cdot 1) = 96.8 \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20 \dots 25) \cdot k}, \text{ людино-годин. (3.5)}$$

Підставивши дані значення в формулу (3.5), отримаємо:

$$t_n = 1936.44 / (25 \cdot 1) = 77.4 \text{ людино-годин}$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

$$t_{oml} = \frac{Q}{(4..5) \cdot k}, \text{ людино-годин. (3.6)}$$

Підставивши дані значення в формулу (3.6), отримаємо:

$$t_{oml} = 1936.44 / (5 \cdot 1) = 387.3 \text{ людино-годин.}$$

за умови комплексного налагодження завдання:

$$t_{oml}^k = 1,5 \cdot t_{oml}, \text{ людино-годин. (3.7)}$$

Підставивши дані значення в формулу (3.7), отримаємо:

$$t_{oml}^k = 1.5 \cdot 387.3 = 580.9 \text{ людино-годин.}$$

Витрати праці на підготовку документації визначаються за формулою:

$$t_{\partial} = t_{\partial p} + t_{\partial o}, \text{ людино-годин, (3.8)}$$

де: $t_{\partial p}$ -трудомісткість підготовки матеріалів і рукопису:

$$t_{\partial p} = \frac{Q}{(15..20) \cdot k}, \text{ людино-годин, (3.9)}$$

$t_{\partial o}$ - трудомісткість редагування, печатки й оформлення документації:

$$t_{\partial o} = 0,75 \cdot t_{\partial p}, \text{ людино-годин. (3.10)}$$

Підставляючи відповідні значення в формули, отримаємо:

$$t_{\partial p} = 1936.44 / (16 \cdot 1.3) = 121.02 \text{ людино-годин.}$$

$$t_{\partial o} = 0.75 \cdot 121.02 = 90.8 \text{ людино-годин.}$$

$$t_{\partial} = 121.02 + 90.8 = 211.82 \text{ людино-годин.}$$

Отримавши дані з всіх вище перерахованих формул можемо підрахувати загальну трудомісткість розробки за допомогою формули (3.1).

$$t = 50 + 28.4 + 96.8 + 77.4 + 580.9 + 211.82 = 1045.3 \text{ людино-години.}$$

3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ Кпо включають витрати на заробітну плату виконавця програми Зз/п і витрат машинного часу, необхідного на налагодження програми на ЕОМ:

$$K_{ПО} = Z_{ЗП} + Z_{МВ}, \text{ грн.} \quad (3.11)$$

Заробітна плата виконавців визначається за формулою:

$$Z_{ЗП} = t \cdot C_{ПР}, \text{ грн,} \quad (3.12)$$

де: t - загальна трудомісткість, людино-годин;

$C_{ПР}$ - середня годинна заробітна плата програміста, грн/година

Враховуючу, що середня заробітна плата програміста становить 128 грн/год та підставляючи дані в формулу маємо:

$$Z_{ЗП} = 1045.3 \cdot 113 = 118\,119 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$Z_{МВ} = t_{отл} \cdot C_{мч}, \text{ грн,} \quad (3.13)$$

де: $t_{отл}$ - трудомісткість налагодження програми на ЕОМ, год.

$C_{мч}$ - вартість машино-години ЕОМ, грн/год.

Підставивши в формулу (3.13) та знаючи, що до вартості ЕОМ також входить вартість покупки ПК маємо:

$$Z_{me} = 580.9 \cdot 20 = 11618 \text{ грн.}$$

На основі отриманих даних витрати на створення програмного продукту наступні:

$$K_{ПО} = 118\,119 + 11618 = 129737 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс. (3.14)}$$

де: B_k - число виконавців (дорівнює 1);

F_p - місячний фонд робочого часу (при 40 годинному робочому тижні $F_p=176$ годин).

Підставляючи дані у формулу (3.14) маємо:

$$T = 1045.3 / (1 \cdot 176) \approx 5.9 \text{ міс.}$$

Висновок: на розробку програмного продукту, що призначений для ознайомлення користувачів з інформаційною системою, необхідно 1045.3 людино-години. З цих даних маємо, що ймовірна тривалість розробки продукту складатиме 5.9 місяці при стандартному робочому тижні та робочому місяці. Вартість даного програмного продукту становитиме 129 737грн. В дану суму входить заробітна плата фахівця та затрати на ЕОМ, а також його покупку.

ВИСНОВКИ

Головною метою кваліфікаційної роботи було розроблення інформаційного веб-додатку для італійського ресторану з можливістю користувачам бронювати столики у ресторані.

Саме цей інформаційний веб-додаток був створений за допомогою наступних технолонії:

- HTML — мова розмітки;
- CSS — каскадні таблиці;
- BEM — популярна конвенція щодо найменування імен класів у HTML і CSS;
- JavaScript — мова програмування.

В результаті веб-додаток має змогу приймати інформацію від потенційних клієнтів щодо бронювання столиків в ресторані, має детальну інформацію про історію та існування ресторану, також е можливість подивитись меню та контакти ресторану.

Завдяки кваліфікаційній роботі були закріплені теоретичні та практичні навички з програмування. Також були досягнені усі цілі, які були поставлені на початку роботи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Стоян Стефанов. JavaScript. Шаблини. - Символ-Плюс, 2018. – 272с
2. JavaScript: Подробное руководство (Definitive Guide). / D. Flanagan.
- Издательство: Символ-Плюс, 2008. – 992 с.
3. HTML и XHTML. Подробное руководство. Шестое издание. / Ч. Муссиано, Б. Кеннеди. – Издательство: Символ-Плюс, 2011. – 752 с.
4. Стивен Шафер. HTML, XHTML и CSS. Библия пользователя, 5-е издание. HTML, XHTML, and CSS Bible, 5th Edition. — М.: «Диалектика», 2010. — 656 с
5. en Frain. Responsive Web Design with HTML5 and CSS: Develop future proof responsive websites using the latest HTML5 and CSS techniques, 3rd Edition Packt Publishing, 2020. - 410p.
6. Jessica Minnick. Responsive Web Design with HTML 5 & CSS (MindTap Course List) 9th Edition. 2020. – 640p
7. VSCode Extensions «VSCode розширення». [Електронний ресурс]. Режим доступу: <https://blog.bitsrc.io/top-10-visual-studio-code-extensions-for-web-developers-5bd6a76bdf5f>
8. Холл М., Браун Л. Программирование для Web.- М.: Издательский дом «Вильямс», 2002.- 1259 с.
9. Chong Lip Phang. Mastering Front-End Web Development: 14 Books in 1. Introducing 200+ Extensions. An Advanced Guide. 2020. – 457
10. НЕ ЗМУШУЙТЕ МЕНЕ ДУМАТИ. Стів Круг. 2000.- 256с.
11. Дизайн звичних речей, Норман Дональд А., 2013.
12. Методичні вказівки з виконання економічного розділу в дипломних проектах студентів спеціальності “Комп’ютерні системи ” / Укладачі О.Г. Вагонова, Нікітіна О.Б. Н.Н. Романюк – Дніпропетровськ: Національний гірничий університет. – 2013. – 23с
13. Сперн В. Figma. Официальный учебный курс / Василий Сперн., 2009. – 512 с. – (ISBN 978-5-699-36200-4, ISBN 978-0-321-57378-0).

14. Shields W. SQL QuickStart Guide: The Simplified Beginner's Guide to Managing, Analyzing, and Manipulating Data With SQL / Walter Shields., 2019. – 251 с.
15. Farrington T. UX Design 2020: The Ultimate Beginner's Guide to User Experience / T. Farrington, T. Brooks, L. Ferrante., 2020.
16. McKay E. UI is Communication: How to Design Intuitive, User Centered Interfaces by Focusing on Effective Communication 1st Edition / Everett McKay., 2013. – 378 с.
17. Гаевский А.Ю. 100% самоучитель. Создание Web-страниц и Webсайтов. HTML и JavaScript / А.Ю. Гаевский, В.А. Романовский. - М.: Триумф, 2014. - 464 с
18. Work.ua «пошук інформації по заробітній платі». [Електронний ресурс]. Режим доступу: <https://work.ua>

ДОДАТОК А

КОД ПРОГРАМИ

Index.html

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

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <link rel="stylesheet" href="css/style.css">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="https://unpkg.com/swiper@8/swiper-bundle.min.css"
/>
  <title>Lucretia</title>
</head>

<body>
  <header class="Header">
    <div class="Header__block">
      <div class="Header__block_logo">
        
      </div>
      <nav class="Header__block_menu">
        <a href="Menu.pdf">Menu</a>
        <a href="5_project_page2/index2.html">About Us</a>
      </nav>
    </div>
  </header>
</body>
</html>
```



```

        <a href="5_project_page3/index3.html">Contacts</a>
    </nav>
    <div class="Header__block_lang">
        <a href="#">Ua</a>
        <a href="#">Eng</a>
    </div>
</div>
</header>

<div class="FirstScreen">

    <div class="FirstScreen__block">
        <div class="FirstScreen__block_title">
            <h1>
                Italian restaurant in the very center of the Dnipro
            </h1>
        </div>

        <div class="FirstScreen__block_button">
            <a href="#" id="open_pop_up">Reserve</a>
        </div>
    </div>

</div>

<div class="pop_up" id="pop_up">
    <div class="pop_up__container">
        <div class="pop_up__body" id="pop_up__body">
            <p>Reserve</p>
            <form action="">
                <input type="text" placeholder="Name">
                <input type="text" placeholder="Phone">
                <button>Send</button>
            </form>
            <div class="pop_up__close" id="pop_up__close">&#10006</div>
        </div>
    </div>
</div>

<div class="Block2">
    <div class="Block2__content">
        <p>
            Lucretia – it is the cuisine and traditions of a family that has
            been <span>inspired by Italy and</span>
            shares his love for food and wine.
        </p>
    </div>
</div>

<div class="Block3">
    <div class="Block3__Content">

```

```
<div class="Block3__Content_img">
  
</div>
```

```
<div class="Block3__Content_text">
  <h2>Restaurant</h2>
  <p>
```

The Lucretia restaurant has 130 seats. The interior, designed by Krivtsova&Redina, is an impeccable ironic eclecticism, where sparkling glossy late-Renaissance frescoed panels coexist with ultra-modern furniture and decorative elements.

```
</p>
<div class="Block3__Content_text_button">
  <a href="5_project_page2/index2.html">more</a>
</div>
</div>
```

```
</div>
</div>
```

```
<div class="Block4">
```

```
  <div class="Block4__content">
    <div class="Block4__content_img1">
      
    </div>
```

Seasonal local products, original author's recipes and gastronomic experiments on the Italian theme are the basis of the menu of the Lucretia restaurant.

```
</p>
<p>
```

Each dish is modern cooking techniques combined with the most reverent attitude to classical culinary traditions.

```
</p>
<div class="Block4__content_text_button">
  <a href="#">see menu</a>
</div>
```

```
</div>
<div class="Block4__content_img2">
  
</div>
<div class="Block4__content_img3">
  
</div>
```

```
</div>
</div>
```

```

<div class="Block5">
  <div class="Block5__content">
    <div class="Block5__content_text">
      <h2>Taste the taste of Italy</h2>
      <p>
        You can book a table in our restaurant <br> any convenient time
or order online home delivery <br>
        on the
        website or by phone.
      </p>
    </div>
    <div class="Block5__content_button">
      <a href="#" id="open_pop_up">Reserve a table</a>
    </div>
  </div>
</div>

<div class="page__gallery gallery">
  <div class="gallery__container">
    <h2 class="gallery__title title">Gallery</h2>
    <div class="gallery__swiper swiper-gallery swiper">
      <!-- Additional required wrapper -->
      <div class="swiper-gallery__wrapper swiper-wrapper">
        <!-- Slides -->
        <div class="swiper-gallery__slide slide-swiper-gallery swiper-
slide">
          <div class="slide-swiper-gallery__picture">
            <div class="slide-swiper-gallery__image">
              
            </div>
          </div>
        </div>
        <div class="swiper-gallery__slide slide-swiper-gallery swiper-
slide">
          <div class="slide-swiper-gallery__picture">
            <div class="slide-swiper-gallery__image">
              
            </div>
          </div>
        </div>
        <div class="swiper-gallery__slide slide-swiper-gallery swiper-
slide">
          <div class="slide-swiper-gallery__picture">
            <div class="slide-swiper-gallery__image">
              
            </div>
          </div>
        </div>
      </div>
      <!-- If we need pagination -->

```

```

                <div class="swiper-gallery__button-prev swiper-button-prev"></div>
                <div class="swiper-gallery__button-next swiper-button-next"></div>
            </div>
        </div>
    </div>

<script src="https://unpkg.com/swiper@8/swiper-bundle.min.js"></script>
<script src="js/script.js"></script>

<div class="footer">
    <div class="footer__container">
        <div class="footer__container_logo">
            
        </div>
        <nav class="footer__container_menu">
            <a href="#">Menu</a>
            <a href="#">About Us</a>
            <a href="#">Contacts</a>
        </nav>
        <div class="footer__container_info">
            <a href="#">Dnipro, Sicheslavskaya nab. 35</a>
            <a href="#">+38 098 555 010</a>
            <a href="#">info@lucretia.com</a>
        </div>

        <div class="footer__container_time">
            Work time: <br> 10:00 – 00:00
        </div>

        <div class="footer__container_button">
            <a href="#" id="open_pop_up">Reserve</a>
        </div>

        <div class="footer__container_rights">
            © 2022 • All rights reserved
        </div>
    </div>
</div>
</body>

</html>

```

Style.css

```

* {
    padding: 0;
    margin: 0;
    border: 0;
}

```

```

}

*,
*:before,
*:after {
    -moz-box-sizing: border-box;
    -webkit-box-sizing: border-box;
    box-sizing: border-box;
}

:focus,
:active {
    outline: none;
}

a:focus,
a:active {
    outline: none;
}

nav,
footer,
header,
aside {
    display: block;
}

html,
body {
    height: 100%;
    width: 100%;
    font-size: 100%;
    line-height: 1;
    font-size: 14px;
    -ms-text-size-adjust: 100%;
    -moz-text-size-adjust: 100%;
    -webkit-text-size-adjust: 100%;
}

input,
button,
textarea {
    font-family: inherit;
}

input::-ms-clear {
    display: none;
}

button {
    cursor: pointer;
}

```

```

}

button::-moz-focus-inner {
    padding: 0;
    border: 0;
}

a,
a:visited {
    text-decoration: none;
}

a:hover {
    text-decoration: none;
}

ul li {
    list-style: none;
}

img {
    vertical-align: top;
}

h1,
h2,
h3,
h4,
h5,
h6 {
    font-size: inherit;
    font-weight: 400;
}

/*-----*/
* {
    font-family: Verdana, Geneva, Tahoma, sans-serif;
}

body {
    background: #262626;
    height: 5000px;
}

.Header {
    width: 100%;
    font-size: 16px;
    /*position: fixed;*/
    position: relative;
    z-index: 1;
}

```

```

.Header__block {
    width: 1620px;
    height: 47px;
    margin: 0 auto;
    position: relative;
    top: 20px;
}

.Header__block_logo {
    display: inline-block;
    position: absolute;
    bottom: 0;
    left: 0;
}

.Header__block_menu {
    text-align: center;
    display: inline-block;
    position: absolute;
    bottom: 0;
    right: 665px;
}

.Header__block_menu a {
    color: #fff;
    margin: 0 0 0 50px;
}

.Header__block_menu a:first-child {
    margin: 0 0 0 0;
}

.Header__block_menu a:hover {
    color: rgb(148, 148, 148);
    padding: 0 0 10px 0;
    border-bottom: 1px solid rgb(148, 148, 148);
}

.Header__block_lang {
    text-align: right;
    display: inline-block;
    position: absolute;
    bottom: 0;
    right: 0;
}

.Header__block_lang a {
    color: #fff;
    margin: 0 0 0 50px;
}

```

```

.Header__block_lang a:last-child {
    color: rgb(148, 148, 148);
}

.FirstScreen {
    width: 100%;
    height: 940px;
    background: url("../img/image_1.png");
    position: absolute;
    top: 0px;
}

.FirstScreen__block {
    color: #fff;
    /*position: absolute;
    top: 350px;
    left: 570px;*/
    margin: 350px auto;
}

.FirstScreen__block_subtitle {
    font-size: 18px;
    text-align: center;
    margin: 0 0 20px 0;
}

.FirstScreen__block_title {
    font-size: 40px;
    text-align: center;
}

.FirstScreen__block_button {
    margin: 50px 0 0 0;
    text-align: center;
}

.FirstScreen__block_button a {
    display: inline-block;
    font-weight: bold;
    border: 1px solid #fff;
    padding: 19px 62px;
    color: #fff;
}

.FirstScreen__block_button a:hover {
    background: #fff;
    color: #000;
    font-weight: bold;
}

```



```
.pop_up {  
  display: none;  
  width: 100%;  
  height: 100%;  
  position: fixed;  
  left: 0;  
  top: 0;  
  background-color: rgba(0, 0, 0, .8);  
  z-index: 2;  
}
```

```
.pop_up.active {  
  display: block;  
}
```

```
.pop_up__container {  
  display: flex;  
  width: 100%;  
  height: 100%;  
}
```

```
.pop_up__body {  
  margin: auto;  
  width: 500px;  
  background-color: #fff;  
  text-align: center;  
  padding: 100px 15px 110px 15px;  
  position: relative;  
}
```

```
.pop_up__body p {  
  font-size: 24px;  
  font-weight: 600;  
  color: #262626;  
  margin-bottom: 40px;  
}
```

```
.pop_up__body input {  
  display: block;  
  margin: 25px auto 0 auto;  
  width: 330px;  
  padding: 17px 20px;  
  background-color: #E5E5E5;  
  border: none;  
  font-weight: 500;  
  font-size: 18px;  
}
```

```
.pop_up__body button {  
  cursor: pointer;  
  display: block;
```

```

width: 330px;
margin: 60px auto 0 auto;
padding: 20px 0;
font-weight: 500;
font-size: 16px;
border: none;
color: #fff;
background: linear-gradient(100.25deg, #200504 0%, #9E5956 144.77%);
}

.pop_up__close {
position: absolute;
top: 15px;
right: 15px;
font-size: 21px;
cursor: pointer;
}

.Block2 {
width: 100%;
height: 450px;
position: absolute;
top: 940px;
font-size: 36px;
line-height: 140%;
color: #fff;
text-align: center;
background: linear-gradient(100.25deg, #200504 0%, #9E5956 144.77%);
}

.Block2::before {
content: url("../img/Star2.svg");
position: absolute;
top: -50px;
left: 910px;
}

.Block2__content {
font-size: 24px;
font-weight: 200;
width: 800px;
margin: 150px auto;
}

.Block2__content span {
font-style: italic;
}

.Block3 {
width: 100%;
position: absolute;

```

```

    top: 1390px;
}

.Block3__Content {
    color: #fff;
}

.Block3__Content_img {
    display: inline;
}

.Block3__Content_img img {
    width: 943px;
    height: 900px;
}

.Block3__Content_text {
    display: inline-block;
    width: 400px;
    font-size: 16px;
    margin: 200px 0 0 113px;
}

.Block3__Content_text h2 {
    font-size: 54px;
}

.Block3__Content_text p {
    margin: 30px 0 0 0;
    line-height: 140%;
}

.Block3__Content_text_button {
    margin: 50px 0 0 0;
    font-size: 16px;
}

.Block3__Content_text_button a {
    margin: 0 0 0 10px;
    color: #fff;
    text-transform: uppercase;
}

.Block3__Content_text_button:before {
    content: url('../img/button.svg');
    position: relative;
    top: 3px;
}

.Block3__Content_text_button:hover:before {
    content: url('../img/button2.svg');
}

```

```

}

.Block4 {
  position: absolute;
  top: 2290px;
}

.Block4__content {
  display: inline-block;
}

.Block4__content_img1 {
  display: inline;
  font-size: 0;
}

.Block4__content_text {
  width: 380px;
  display: inline-block;
  color: #fff;
  margin: 200px 39px 0 39px;
}

.Block4__content_text h2 {
  font-size: 54px;
  margin: 0 0 30px 0;
}

.Block4__content_text p {
  font-size: 16px;
  line-height: 140%;
  margin: 0 0 30px 0;
}

.Block4__content_img2 {
  display: inline;
  font-size: 0;
}

.Block4__content_img3 {
  display: inline;
  font-size: 0;
}

.Block4__content_text_button {
  margin: 50px 0 0 0;
  font-size: 16px;
}

```

```

.Block4__content_text_button a {
  margin: 0 0 0 10px;
  color: #fff;
  text-transform: uppercase;
}

.Block4__content_text_button:before {
  content: url('../img/button.svg');
  position: relative;
  top: 3px;
}

.Block4__content_text_button:hover:before {
  content: url('../img/button2.svg');
}

.Block5 {
  width: 100%;
  height: 420px;
  position: absolute;
  top: 3100px;
  font-size: 36px;
  line-height: 140%;
  color: #fff;
  text-align: center;
  background: linear-gradient(100.25deg, #200504 0%, #9E5956 144.77%);
}

.Block5__content {
  margin: 65px auto;
}

.Block5__content_text p {
  font-size: 16px;
  text-align: center;
  line-height: 140%;
  padding: 30px 0 0 0;
  margin: 0 0 50px 0;
}

.Block5__content_button {
  position: absolute;
  left: 830px;
}

.Block5__content_button a {
  display: inline-block;
  font-weight: bold;
  border: 1px solid #fff;
}

```

```

padding: 10px 60px;
color: #fff;
font-size: 14px;
}

.Block5__content_button a:hover {
background: #fff;
color: #000;
font-weight: bold;
}

.gallery {
position: absolute;
width: 100%;
top: 3640px;
text-align: center;
}

.title {
color: #fff;
margin-bottom: 50px;
font-size: 54px;
}

.swiper-button-prev {
margin-left: 130px;
color: #fff;
}

.swiper-button-next {
margin-right: 130px;
color: #fff;
}

.footer {
position: absolute;
width: 100%;
height: 400px;
top: 4696px;
background: linear-gradient(100.25deg, #200504 0%, #9E5956 144.77%);
}

.footer__container {
margin: 100px 0 0 350px;
}

.footer__container_menu {
position: absolute;
left: 650px;
top: 100px;
}

```

```

    display: flex;
    flex-direction: column;
    align-items: flex-start;
    padding: 0px;
    gap: 30px;
}

.footer__container_menu a {
    color: #fff;
}

.footer__container_info {
    display: flex;
    flex-direction: column;
    align-items: flex-start;
    padding: 0px;
    gap: 30px;
    position: absolute;
    left: 900px;
    top: 100px;
}

.footer__container_info a {
    color: #fff;
}

.footer__container_time {
    display: flex;
    flex-direction: column;
    align-items: flex-start;
    position: absolute;
    left: 1300px;
    top: 100px;
    color: #fff;
    line-height: 140%;
}

.footer__container_button {
    display: flex;
    flex-direction: column;
    align-items: flex-start;
    position: absolute;
    left: 1300px;
    top: 160px;
    margin: 0px 0 0 0;
    text-align: center;
}

.footer__container_button a {
    display: inline-block;
    font-weight: bold;
}

```

```

border: 1px solid #fff;
padding: 19px 62px;
color: #fff;
}

.footer__container_button a:hover {
background: #fff;
color: #000;
font-weight: bold;
}

.footer__container_rights {
position: absolute;
left: 845px;
top: 350px;
color: #A6908F;
}

```

}

Script.js

```
"use strict"
```

```
const gallerySwiper = document.querySelector('.swiper-gallery');
```

```
if (gallerySwiper) {
  const swiper = new Swiper('.swiper-gallery', {
    // Optional parameters
    autoHeight: true,
    loop: true,
    // Navigation arrows
    navigation: {
      nextEl: '.swiper-button-next',
      prevEl: '.swiper-button-prev',
      clickable: true
    },
  });
}

```

```
const openPopUp = document.getElementById('open_pop_up')
const closePopUp = document.getElementById('pop_up__close')
const popUp = document.getElementById('pop_up')
```

```
openPopUp.addEventListener('click', function (e) {
  e.preventDefault();
  popUp.classList.add('active');
})

```

```
closePopUp.addEventListener('click', () => {
  popUp.classList.remove('active');
})

```


Index.html2

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

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <link rel="stylesheet" href="css/style.css">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="https://unpkg.com/swiper@8/swiper-bundle.min.css"
/>
  <title>Lucretia</title>
</head>

<body>
  <header class="Header">
    <div class="Header__block">
      <div class="Header__block_logo">
        
      </div>
      <nav class="Header__block_menu">
        <a href="#">Menu</a>
        <a href="#">About Us</a>
        <a href="Pages/index3.html">Contacts</a>
      </nav>
      <div class="Header__block_lang">
        <a href="#">Ua</a>
        <a href="#">Eng</a>
      </div>
    </div>
  </header>

  <div class="firstscreen">
    <div class="firstscreen__text">
      <h2>About Us</h2>
    </div>
  </div>

  <div class="founder">
    <div class="founder__img"></div>
    <div class="founder__text">
      <h3>Gino De Luca</h3>
      <h4>founder of Lucretia</h4>
      <p>«Italian cuisine is limitless for me, as it is very unique in
different regions, from the Italian Alps to
the island of Sicily. Representing Italian cuisine in Ukraine, for
me, as for every Italian, it is
```

important to put love into creating a gastronomic masterpiece so that the guest receives a freshly prepared dish in a modern and appetizing serving of quality products.»</p>

```
</div>
</div>
```

```
<div class="foundation">
  <div class="foundation__img"></div>
  <div class="foundation__text">
    <h3>Foundation</h3>
    <p>Lucretia – legendary Italian restaurant, founded in 2003, with an interesting long history, high-class service, exquisite Italian cuisine with Mediterranean motives, fresh seafood and the best meat in the world.</p>
```

```
</div>
</div>
```

```
<div class="interior">
  <div class="interior__img"></div>
  <div class="interior__text">
    <h3>Interior</h3>
    <p>The Lucretia restaurant has 130 seats. The interior was designed by Krivtsova&Redina and is an impeccable ironic eclecticism, where sparkling glossy panels with late renaissance frescoes coexist with ultra-modern furniture and decorative elements.</p>
```

```
</div>
</div>
```

```
<div class="footer">
  <div class="footer__container">
    <div class="footer__container_logo">
      
    </div>
    <nav class="footer__container_menu">
      <a href="#">Menu</a>
      <a href="#">About Us</a>
      <a href="#">Contacts</a>
    </nav>
    <div class="footer__container_info">
      <a href="#">Dnipro, Sicheslavskaya nab. 35</a>
      <a href="#">+38 098 555 010</a>
      <a href="#">info@lucretia.com</a>
    </div>
```

```
<div class="footer__container_time">
  Work time: <br> 10:00 – 00:00
</div>
```

```
        <div class="footer__container_button">
            <a href="#">Reserve</a>
        </div>

        <div class="footer__container_rights">
            © 2022 • All rights reserved
        </div>
    </div>
</div>
</body>

</html>
```

Style.css

```
* {
    padding: 0;
    margin: 0;
    border: 0;
}

*,
*:before,
*:after {
    -moz-box-sizing: border-box;
    -webkit-box-sizing: border-box;
    box-sizing: border-box;
}

:focus,
:active {
    outline: none;
}

a:focus,
a:active {
    outline: none;
}

nav,
footer,
header,
aside {
    display: block;
}
```

```
html,
body {
    height: 100%;
    width: 100%;
    font-size: 100%;
    line-height: 1;
    font-size: 14px;
    -ms-text-size-adjust: 100%;
    -moz-text-size-adjust: 100%;
    -webkit-text-size-adjust: 100%;
}
```

```
input,
button,
textarea {
    font-family: inherit;
}
```

```
input::-ms-clear {
    display: none;
}
```

```
button {
    cursor: pointer;
}
```

```
button::-moz-focus-inner {
    padding: 0;
    border: 0;
}
```

```
a,
a:visited {
    text-decoration: none;
}
```

```
a:hover {
    text-decoration: none;
}
```

```
ul li {
    list-style: none;
}
```

```
img {
    vertical-align: top;
}
```

```
h1,
h2,
```

```

h3,
h4,
h5,
h6 {
    font-size: inherit;
    font-weight: 400;
}

/*-----*/
* {
    font-family: Verdana, Geneva, Tahoma, sans-serif;
}

body {
    width: 192;
    background: #262626;
}

.Header {
    width: 100%;
    font-size: 16px;
    /*position: fixed;*/
    position: relative;
    z-index: 1;
}

.Header__block {
    width: 1620px;
    height: 47px;
    margin: 0 auto;
    position: relative;
    top: 20px;
}

.Header__block_logo {
    display: inline-block;
    position: absolute;
    bottom: 0;
    left: 0;
}

.Header__block_menu {
    text-align: center;
    display: inline-block;
    position: absolute;
    bottom: 0;
    right: 665px;
}

.Header__block_menu a {
    color: #fff;
}

```

```

    margin: 0 0 0 50px;
}

.Header__block_menu a:first-child {
    margin: 0 0 0 0;
}

.Header__block_menu a:hover {
    color: rgb(148, 148, 148);
    padding: 0 0 10px 0;
    border-bottom: 1px solid rgb(148, 148, 148);
}

.Header__block_menu a:nth-child(2) {
    color: rgb(148, 148, 148);
}

.Header__block_menu a:nth-child(2):hover {
    color: rgb(148, 148, 148);
    padding: 0;
    border-bottom: none;
}

.Header__block_lang {
    text-align: right;
    display: inline-block;
    position: absolute;
    bottom: 0;
    right: 0;
}

.Header__block_lang a {
    color: #fff;
    margin: 0 0 0 50px;
}

.Header__block_lang a:last-child {
    color: rgb(148, 148, 148);
}

.firstscreen {
    position: absolute;
    background: url("../img/FirstScreen.png");
    width: 100%;
    height: 481px;
    left: 0px;
    top: 0px;
}

.firstscreen__text h2 {
    position: absolute;
}

```

```
    left: 800px;
    top: 280px;
    font-size: 64px;
    color: #fff;
}

.founder {
    position: absolute;
    height: 800px;
    left: 0px;
    top: 481px;
}

.founder__img {
    margin: 0 0 0 350px;
}

.founder__text {
    position: absolute;
    width: 495px;
    height: 450px;
    left: 1073px;
    top: 248px;
    color: #fff;
    font-size: 16px;
}

.founder__text h3 {
    font-size: 54px;
}

.founder__text h4 {
    margin-top: 10px;
    font-size: 24px;
    font-style: italic;
    color: #949494;
}

.founder__text p {
    line-height: 140%;
    margin-top: 30px;
}

.foundation {
    position: absolute;
    height: 650px;
    left: 0px;
    top: 1281px;
}
```

```
.foundation__img {  
  position: absolute;  
  left: 960px;  
  top: 0px;  
}
```

```
.foundation__text {  
  
  position: absolute;  
  width: 495px;  
  height: 201px;  
  left: 352px;  
  top: 224px;  
  color: #fff;  
  font-size: 16px;  
}
```

```
.foundation__text h3 {  
  font-size: 54px;  
}
```

```
.foundation__text p {  
  line-height: 140%;  
  margin-top: 30px;  
}
```

```
.interior {  
  position: absolute;  
  height: 650px;  
  top: 1931px;  
}
```

```
.interior__img {  
  position: absolute;  
  left: 352px;  
  top: 0px;  
}
```

```
.interior__text {  
  position: absolute;  
  height: 226px;  
  left: 1073px;  
  top: 200px;  
  color: #fff;  
}
```

```
.interior__text h3 {  
  font-size: 54px;  
}
```

```
.interior__text p {
```



```

width: 495px;
margin-top: 30px;
font-size: 16px;
line-height: 140%;
}

.footer {
  position: absolute;
  width: 100%;
  height: 400px;
  top: 2781px;
  background: linear-gradient(100.25deg, #200504 0%, #9E5956 144.77%);
}

.footer__container {
  margin: 100px 0 0 350px;
}

.footer__container_menu {
  position: absolute;
  left: 650px;
  top: 100px;
  display: flex;
  flex-direction: column;
  align-items: flex-start;
  padding: 0px;
  gap: 30px;
}

.footer__container_menu a {
  color: #fff;
}

.footer__container_info {
  display: flex;
  flex-direction: column;
  align-items: flex-start;
  padding: 0px;
  gap: 30px;
  position: absolute;
  left: 900px;
  top: 100px;
}

.footer__container_info a {
  color: #fff;
}

.footer__container_time {
  display: flex;
  flex-direction: column;
}

```

```

    align-items: flex-start;
    position: absolute;
    left: 1300px;
    top: 100px;
    color: #fff;
    line-height: 140%;
}

.footer__container_button {
    display: flex;
    flex-direction: column;
    align-items: flex-start;
    position: absolute;
    left: 1300px;
    top: 160px;
    margin: 0px 0 0 0;
    text-align: center;
}

.footer__container_button a {
    display: inline-block;
    font-weight: bold;
    border: 1px solid #fff;
    padding: 19px 62px;
    color: #fff;
}

.footer__container_button a:hover {
    background: #fff;
    color: #000;
    font-weight: bold;
}

.footer__container_rights {
    position: absolute;
    left: 845px;
    top: 350px;
    color: #A6908F;
}

```

Решта файлів буде прикріплена до роботи носії на якому буде архів з повною роботою.

ДОДАТОК Б

ВІДГУК

керівника економічного розділу
на кваліфікаційну роботу бакалавра

на тему:

«Розробка інформаційного веб-додатку для італійського ресторану»

студента групи 122-18-2 Варезкіна Іллі Олександровича

Керівник економічного розділу
доцент каф. ПЕП та ПУ, к.е.н

Л. В. Касьяненко

ДОДАТОК В**ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ**

| Ім'я файлу | Опис |
|--|--|
| Пояснювальні документи | |
| Кваліфікаційна робота Варежкін.docx | Пояснювальна записка до кваліфікаційної роботи. Документ Word. |
| Кваліфікаційна робота Варежкін.pdf | Пояснювальна записка до кваліфікаційної роботи в форматі PDF |
| Програма | |
| Варежкін кваліфікаційна робота.rar | Архів. Містить коди програми і скомпільовану програму |
| Презентація | |
| Презентація Варежкін.ppt | Презентація кваліфікаційної роботи |