

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 Іванова Дениса Вікторовича
(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 веб-орієнтованої інформаційної системи для
ресторану з використанням фреймворку ReactJS

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц. Приходченко С.Д.			
розділів:				
спеціальний	доц. Приходченко С.Д.			
економічний	доц. Касьяненко Л.В.			
Рецензент				
Нормоконтролер	доц. Гуліна І.Г.			

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)

Іванов Д.В.

(прізвище та ініціали)

тема кваліфікаційної роботи

Розробка веб-орієнтованої інформаційної

Системи для ресторану з використанням фреймворку ReactJS

затверджена наказом ректора НТУ «ДП» від

18.05.2022

№ 268-с

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

Завдання видав

доц. Приходченко С.Д.

(підпис)

(посада, прізвище, ініціали)

Завдання прийняв до виконання

Іванов Д.В.

(підпис)

(прізвище, ініціали)

Дата видачі завдання: 14.01.2022 р.

Термін подання кваліфікаційної роботи до ЕК: 13.06.2022 р.

РЕФЕРАТ

Пояснювальна записка: 63 с., 26 рис., 3 дод., 20 джерела.

Об'єкт розробки: веб-орієнтована інформаційна система.

Мета кваліфікаційної роботи: створення веб-орієнтованої інформаційної системи для більшої взаємодії клієнта з продуктом чи послугою.

У вступі розглядається сучасний стан проблеми, визначається конкретна мета кваліфікаційної роботи та сфера її застосування, наводиться аргументування актуальності теми та деталізується постановка завдання

У першому розділі проаналізовано предметну галузь, визначено актуальність завдання та призначення розробки, сформульовано постановку завдання, зазначено вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі проаналізовані наявні рішення реалізації, обрано платформи для розробки, вибрано інструменти, описана робота програми, сформована структура сайту, створено дизайн продукту.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, проведено розрахунок витрат в часі на створення продукту та підрахунок вартості робочих годин по виготовленню системи.

Практичне завдання полягає у створенні інформаційної системи, що надає можливість клієнту ознайомитися з інформацією про ресторан та переглянути список товарів та послуг закладу.

Актуальність даного продукту визначається великим попитом на дані розробки, що допомагають поширювати та рекламувати продукт.

Список ключових слів: КОМП'ЮТЕР, ІНФОРМАЦІЙНА СИСТЕМА, ДИЗАЙН, АДАПТИВНІСТЬ САЙТУ.

ABSTRACT

Explanatory note: 63 p., 26 pict., 3 apps., 20 sources.

Object of development: web-oriented information system.

The purpose of the qualification work: to create a web-based information system for greater customer interaction with the product or service.

The introduction considers the current state of the problem, determines the specific purpose of the qualification work and its scope, provides arguments for the relevance of the topic and details the task statement

In the first section the subject branch is analyzed, the urgency of the task and the purpose of development are determined, the statement of the task is formulated, the requirements to the software implementation, technologies and software are specified.

The second section analyzes the existing implementation solutions, selects platforms for development, selects tools, describes the work of the program, formed the structure of the site, created a product design.

The economic section determines the complexity of the developed information system, calculates the cost of time to create a product and calculates the cost of working hours for the manufacture of the system.

The practical task is to create an information system that allows the customer to read information about the restaurant and view the list of goods and services of the institution.

The relevance of this product is determined by the high demand for development data that helps to distribute and promote the product.

Keywords: COMPUTER, INFORMATION SYSTEM, DESIGN, ADAPTABILITY OF THE SITE.

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ - програмне забезпечення;

IDE - Integrated development environment;

UURL - Uniform Resource Locator;

HTML - HyperText Markup Language;

TCP - Transmission Control Protocol;

IP - Internet Protocol;

HTTP - HyperText Transfer Protocol.

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
СПИСОК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ	9
1.1. Загальні відомості з предметної галузі	9
1.2. Призначення розробки та галузь застосування.....	11
1.3. Підстави для розробки.....	12
1.4. Постановка завдання.....	12
1.5. Вимоги до програми або програмного виробу	13
1.5.1. Вимоги до функціональних характеристик.....	13
1.5.2. Вимоги до інформаційної безпеки	13
1.5.3. Вимоги до складу та параметрів технічних засобів	13
1.5.4. Вимоги до інформаційної та програмної сумісності.....	14
РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ	15
2.1. Функціональне призначення програми.....	15
2.2. Опис застосованих математичних методів.....	15
2.3. Опис використаних технологій та мов програмування.....	15
2.4. Опис структури системи та алгоритмів її функціонування	19
2.5. Обґрунтування та організація вхідних та вихідних даних програми	24
2.6. Опис розробленої системи.....	25
2.6.1. Використані технічні засоби	26
2.6.2. Використані програмні засоби.....	27
2.6.3. Виклик та завантаження програми	30
2.6.4. Опис інтерфейсу користувача.....	31
РОЗДІЛ 3 ЕКОНОМІЧНИЙ РОЗДІЛ	38
3.1. Розрахунок трудомісткості та вартості розробки програмного продукту .	38
3.2. Розрахунок витрат на створення програми	42
ВИСНОВКИ.....	45

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	46
ДОДАТОК А КОД ПРОГРАМИ.....	48
ДОДАТОК Б ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ	62
ДОДАТОК В Перелік файлів на диску	63

ВСТУП

Завданням даної кваліфікаційної роботи є інформаційна система призначена для застосування у сфері ресторанного бізнесу.

В наш час важко уявити хоча б одне підприємство без інформаційної системи. Сайт – це чудовий інструмент для ведення бізнесу, збору важливої інформації про потенційних клієнтів, залучення нових користувачів та дистанційне управління різними бізнес-процесами.

Інтернет надає змогу до розширити охоплення аудиторії, тому люди по всьому світі можуть дізнатися про ваш бізнес. З сайтом потенційні клієнти завжди можуть дізнатися про вас чи ваш продукт у зручний для себе час. Вони не обмежені можливістю зв'язатися з вами тільки у щоденні години роботи. Якщо ваша компанія закрита у вихідні чи святкові дні, клієнти всеодно мають можливість дізнатися про вас в режимі онлайн.

Багато людей користуються Інтернетом перед покупкою, щоб дізнатися, що їм підходить чи потрібно більше. Якщо користувач не може знайти ваш продукт чи послугу в Інтернеті, то більшість клієнтів підійдуть до вас не так серйозно, як до конкурента з сайтом.

Тому проблема, розглянута в даній кваліфікаційній роботі є актуальною.

Метою кваліфікаційної роботи є вивчення та вдосконалення засобів для створення веб-орієнтованих інформаційних систем. Я вважаю, що набуті навички будуть актуальними у сфері будь-якого бізнесу.

Основними вимогами системи будуть: надійна та стабільна робота, дружній та інтуїтивний інтерфейс та спроможність системи адаптуватися до різних пристроїв.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Загальні відомості з предметної галузі

Обмін інформацією між клієнтом та сервером відбувається за рахунок протоколів. За допомогою протоколу TCP відбувається поділ даних на окремі пакети та їх збирання в пунктах призначення.

Безпосередня ж передача пакетів мережею відбувається за допомогою протоколу IP.

Кожен комп'ютер має IP-адресу, за допомогою неї він має змогу приймати або передавати будь-які дані. Комп'ютер розпізнає та запам'ятовує свою адресу, для звичайного користувача запам'ятати велику кількість цифр складно, тому розробники присвоюють серверам доменні імена. В кожній країні зазвичай доменом першого рівня є код країни.

Тому слід зробити висновок, що система доменних імен має свою структуру та організована в зручний для користувачів спосіб.

Приблизно в 90-х роках минулого століття почалося створення домашніх сторінок. Зараз же кожна людина чи компанія має свою особисту сторінку.

За допомогою найпопулярнішої на даний час службою Інтернету – Веб стало можливим масовий доступ користувачів до мережі. Веб за допомогою протоколу HTTP має змогу обмінюватися даними між серверами. Переглядати ці сторінки користувачі можуть зі спеціальних десктопних чи мобільних програм – браузерів. На сьогодні існує величезна кількість різновидів браузерів на будь-який смак.

Користувачі можуть обмінюватися текстом, ілюстраціями або гіпертекстовими посиланнями. Майже кожен сайт має навігацію – перехід за допомогою гіперпосилань від однієї веб-сторінки до іншої.

Якщо вщяти деяку кількість сторінок яка пов'язана між собою доменним ім'ям, стилями та тематикою, то групу цих сторінок можна назвати веб-сайтом.

Загалом існують мільйони сайтів. І кожен з них має свій унікальний зовнішній вигляд та тематику. Але все ж одна однакова риса присутня, кожен сайт структуру.

Будь-який сайт починається з головної сторінки. Вона займає особливе місце при розробці проекту. Зверху головної сторінки, як правило, розташована шапка. Зазвичай вона дублюється на інших сторінка сайту.

Також одним з найважливіших компонентів сайту є меню. Завдяки правильно скомпонованому меню, потенційні користувачі можуть інтуїтивно переходити по сторінках. Наповнення ж сторінок кожен власник обирає на свій розсуд.

Для того, щоб ваш сайт став доступним для глобального користування, йому слід створити та призначити доменне ім'я. В допомозі з цим питання існують спеціальні організації – хостинги. На сьогодні ви можете користуватися як платними так і безкоштовними хостингами. Вони надають змогу розмістити сайт на своїх серверах та в майбутньому надають його адміністрування.

В 2019 році, через covid-19, виникла нагальна потреба бізнесу у створенні веб-сайтів. Складно уявити яким сферам він не потрібен, тому що даний спосіб просування корисний для всіх.

Після введення карантину користувачі почали частіше купляти в інтернеті. З початку березня 2019 року, кількість пошукових запитів зі словами “онлайн-покупки” чи ”як купляти в інтернеті” по всьому світі виросла вдвічі. А кількість онлайн-покупок в сфері FMCG досягло пікових значень в другій половині березня 2019 року.

Люди все активніше використовують цифрові технології у всіх сферах життя. Як результат, очікування користувачів від онлайн-сервісів та інтернет-

магазинів ростуть. Саме тому всім компаніям, незалежно від їхнього розміру, слід переходити на цифрові технології.

Новий звіт Mckinsey показує, що через пандемію поведінка користувачів змінилася, та більша частина компаній адаптують свої бізнес-моделі під digital.

В минулому році грамотний цифровий маркетинг міг принести до 20% додаткового доходу і скорочення витрат на 30%. За даними BCG, тільки 2% компаній перейшли до більш пізніх етапів цифрової зрілості, останні ж тільки починають цей шлях.

То ж з вищеперерахованого можна зробити висновок, що актуальність даної галузі тільки починає набирати обертів.

1.2. Призначення розробки та галузь застосування

Інформаційна система має назву «Розробка веб-орієнтованої інформаційної системи для ресторану з використанням фреймворку ReactJS».

React - це безкоштовна бібліотека JavaScript із відкритим вихідним кодом для створення інтерфейсів користувача на основі компонентів інтерфейсу користувача. Його підтримують Meta (раніше Facebook) і спільнота окремих розробників і компаній. React можна використовувати як базу для розробки односторінкових, мобільних або серверних додатків із такими фреймворками, як Next.js. Однак React займається лише керуванням станом і відтворенням цього стану в DOM, тому створення додатків React зазвичай вимагає використання додаткових бібліотек для маршрутизації, а також певної функціональності на стороні клієнта.

Розроблена в ході виконання інформаційна система може використовуватися в будь-яких сферах бізнесу. Вона допомагатиме потенційному клієнту швидше ознайомитися з товарами чи послугами. Допомагатиме власнику бізнесу просувати свій продукт.

Метою даної дипломної роботи буде реалізація веб-орієнтованої інформаційної системи, яка буде знайомити користувачів Інтернету з рестораном та його внутрішню інформацією.

1.3. Підстави для 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 веб-орієнтованої інформаційної системи для ресторану з використанням фреймворку ReactJS”.

1.4. Постановка завдання

Завданням даної роботи є створення клієнтської частини веб-орієнтованої інформаційної системи для ресторану, з використанням фреймворку ReactJS.

В результаті слід спроектувати та розробити сайт ресторану.

Кінцевий продукт матиме можливість перегляду інформації про ресторан, інформацію про події та свята проведені в ньому, меню ресторану.

Для реалізації даного проекту було обрано мультипарадигмну мову програмування JavaScript.

Інформаційна система була створена за допомогою фреймворку ReactJS. Дана технологія підходить як для маленьких проектів, так і для гігантів ринку, таких як Facebook.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Кінцевий продукт повинен дотримуватися наступних функціональних характеристик:

- інтуїтивно зрозумілий інтерфейс;
- адаптивність вмісту системи;
- надавати всю необхідну користувачеві інформацію в зручному вигляді.

1.5.2. Вимоги до інформаційної безпеки

Основними вимогами до інформаційної системи є:

- доступність інформації;
- цілісність даних;
- конфіденційність інформації;
- вірогідність даних.

1.5.3. Вимоги до складу та параметрів технічних засобів

Для роботи з веб-орієнтованою інформаційною системою використовується будь-який встановлений браузер. Відповідні характеристики необхідні для роботи більшості браузерів для Windows, Linux чи macOS:

- ЦП: Pentium 4;
- відеоадаптер: 3D адаптер nVidia, Intel, AMD/ATI;
- відеопам'ять: 128 МБ;
- накопичувач: 150 Гб SSD;
- оперативна пам'ять: 2 Гб.

Для стабільної роботи на мобільних пристроях необхідні мінімальні версії операційних систем:

- iOS 11.0 та новіше;
- Android 4.4 та новіше.

1.5.4. Вимоги до інформаційної та програмної сумісності

Основною мовою програмування є JavaScript. Наповнення сайту було розроблене за допомогою бібліотеки React. Для редагування вихідного коду був обраний редактор Visual Studio Code.

Так як інформаційна система є веб системою клієнт взаємодіє з нею за допомогою будь-якого доступного браузера. Тому проблем з завантаження додаткової платформи для взаємодії відсутня.

Для коректної роботи системи слід користуватися такими браузерами:

- Google Chrome від 86 версії;
- Opera від 70 версії;
- Microsoft Edge від 85 версії;

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

2.1. Функціональне призначення програми

Результатом даної кваліфікаційної роботи має бути веб-орієнтована система, що допомагатиме користувачу ознайомитися з інформацією про ресторани.

Система буде запускатися на різних пристроях, так як вона пристосована до браузера, що дає змогу користуватися веб-додатком на будь-яких пристроях та операційних системах.

Основним призначенням системи слугує:

- ознайомлення потенційного клієнту з інформацією про ресторани;
- інформування клієнту про свята та події, що проходять в ресторані;
- перегляд меню.

2.2. Опис застосованих математичних методів

Під час проектування та розробки даної веб-орієнтованої інформаційної системи математичні методи не використовувалися. Використовувалися лише прості арифметичні операції.

2.3. Опис використаних технологій та мов програмування

Дана інформаційна система була розроблена за допомогою наступних технологій:

- React.js;
- React-router-dom;
- JavaScript;
- Node.js;

- CSS3;
- HTML5;

React.js - це фреймворк для створення користувацьких інтерфейсів з відкритим вихідним кодом. Найчастіше React використовується для створення односторінкових чи мобільних додатків.

За допомогою даного фреймворку нам достатньо описати, як частини інтерфейсу додатку виглядають, все інше React зробить за вас (рис. 2.1.).

За рахунок того, що логіка компонентів React написана на мові програмування JavaScript, ми можемо різні дані по системі та тримати стан поза DOM (document object model).

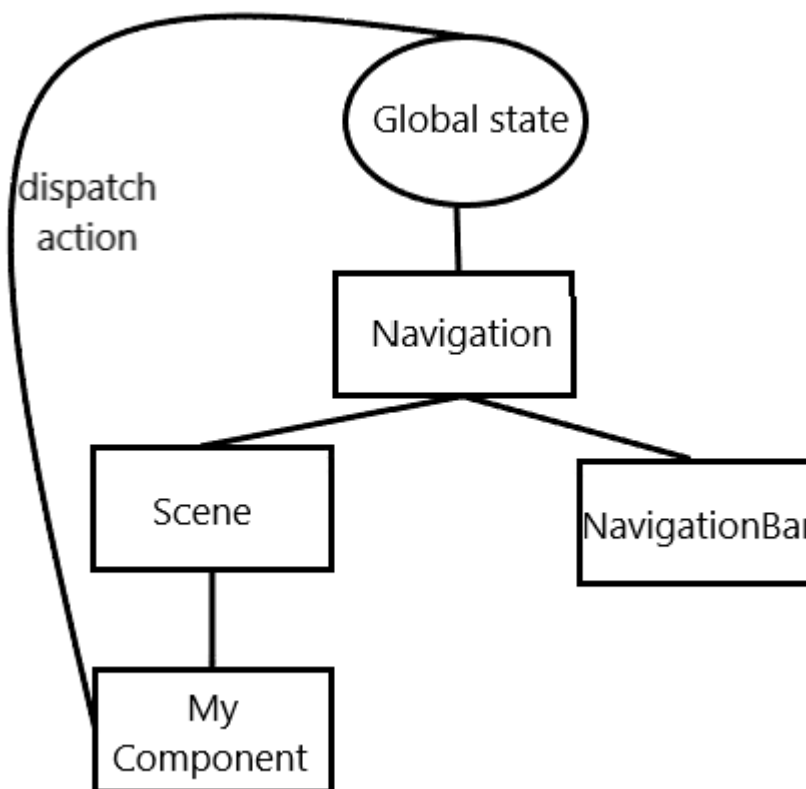


Рис 2.1. – Схема роботи React

React-router-dom – це стандартна бібліотека для маршрутизації в React додатках. Вона є однією з найвикористовуваних бібліотек сьогодні.

В React існує своя система маршрутизації, що дозволяє порівнювати створені компоненти з запитами до системи.

React Router дозволяє маршрутизувати потоки даних в нашій системі зрозумілим чином. За допомогою нього налаштовується правильний роутинг посилянь та завантаження різних сторінок (рис. 2.2.).

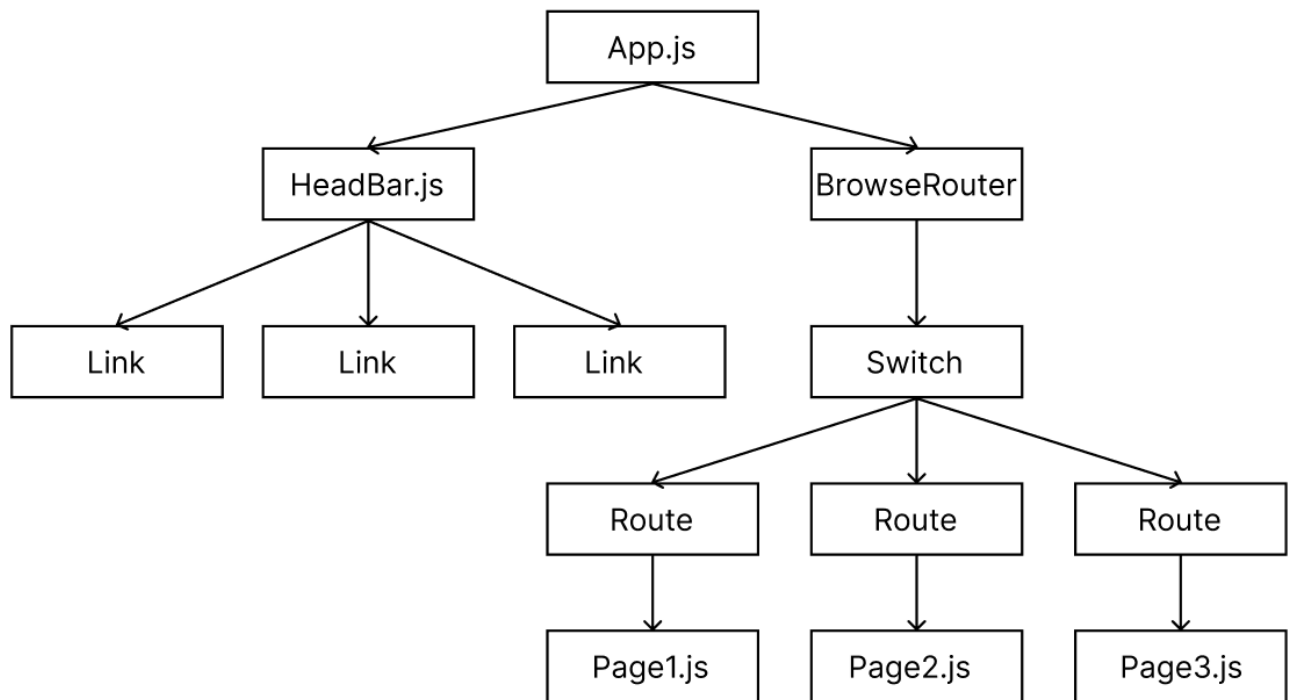


Рис 2.2. – Схема роботи маршрутизації в React

JavaScript – мультипарадигмна мова програмування. Майбільшого використання дана мова набула в браузерах. За допомогою даної мови програмісти надають інтерактивності веб-сторінкам.

JavaScript являє собою об'єктно-орієнтовану мову.

При використанні мови JavaScript я відмітив основні переваги даної технології:

- простота в використанні;
- швидкість роботи в браузері;
- універсальність та поєднання з Node.js;
- систематичне оновлення та поліпшення JavaScript.

Node.js – це середовище виконання JavaScript. За допомогою Node.js ми можемо створювати невеликі повноцінні додатки без великої інфраструктури. Також він дає можливість масштабувати код.

Окрім всього того функціоналу, що надає нам Node.js, в його склад входить незамінна компонента – npm (node package manager).

За допомогою даного засобу ми можемо взаємодіяти з онлайн платформами, тобто браузерними чи серверними. Також ця утиліта дає можливість встановлювати та удаляти пакети, управляти версіями та залежностями, що потрібні для запуску та роботи нашого проекту.

HTML – це стандартизована мова розмітки веб-сторінок та веб-додаків. Всі браузери отримують документ типу HTML від сервера за допомогою протоколів прикладного рівня передачі даних HTTP або HTTPS.

Всі теги HTML документу являють собою блоки. З цих блоків і формується документ. Існує велика кількість тегів, більшість з них є обгорткою для наповнення та можуть включати в себе безмежну кількість інших тегів. Деякі ж з них, такі як `` або `<input />`, виводять вміст на сторінку напряму.

HTML5 – це найновітніша версія мови розмітки гіпертексту. HTML5 був розроблений з пропозиціями та побажаннями програмістів, тому в ньому є майже все для роботи в Інтернеті.

Зі свого створення він сильно еволюціонував. На початку він ледве міг відтворювати та обробляти простий макет сторінки, зараз же можна писати програми, що працюють навіть без підключення до мережі, обробляти відео найбільшої якості та надавати сучасним сайтам надзвичайної графіки.

CSS – це формальна мова для опису зовнішнього вигляду сторінок в веб-додатку. Вона тісно пов'язана з мовою розмітки HTML.

Основним завданням CSS є відгородження логічної структури веб-сторінки та опис зовнішнього вигляду цієї ж сторінки.

Завдяки CSS ми можемо один раз описати стилі для елементів сторінки, а потім просто підключити всі потрібні нам файли для стилізації. Це економить розробнику час та дає можливість писати менше коду в HTML-файлі. Що в свою чергу надає більшої швидкості для завантаження сторінки. Так як браузеру доведеться обробляти меншу кількість коду.

Насьогодні використання атрибутів HTML застаріло, тому більшість програмістів рекомендує використовувати CSS на всіх веб-сторінках. Це дасть можливість зробити їх сумісними з майбутніми браузерами.

CSS3 – це каскадні таблиці стилів третього покоління. Основною перевагою CSS3 від CSS є наявність модулів.

CSS модулі – файл CSS, в якому об'являється локально назва класів. Також він підтримує адаптивний дизайн.

2.4. Опис структури системи та алгоритмів її функціонування

Для проектування та розробки системи необхідно, перш за все, визначити тип структури сайту, який нам необхідний.

Можна виділити чотири основні типи структур веб-сайтів:

- стандартна;
- каскад;
- хмарочос;
- павутина.

Кожен тип має свої переваги та недоліки.

Стандартний тип характеризується тим, що головна сторінка має посилання на інші сторінки, а вони, в свою чергу, мають посилання на головну сторінку сайту.

Каскад характеризується одним напрямком переміщення по веб-сторінкам сайту. Тобто існує тільки один шлях для переходу по сторінках.

Хмарочос характеризується тим, що опинитися на деяких веб-сторінках можливо тільки слідуєчи правильній маршрутизації.

Павутина характеризується здатністю переходу з практично будь-якої сторінки на будь-яку.

При проектуванні інформаційної системи я обрав тип структури системи хмарочос (рис. 2.3.).

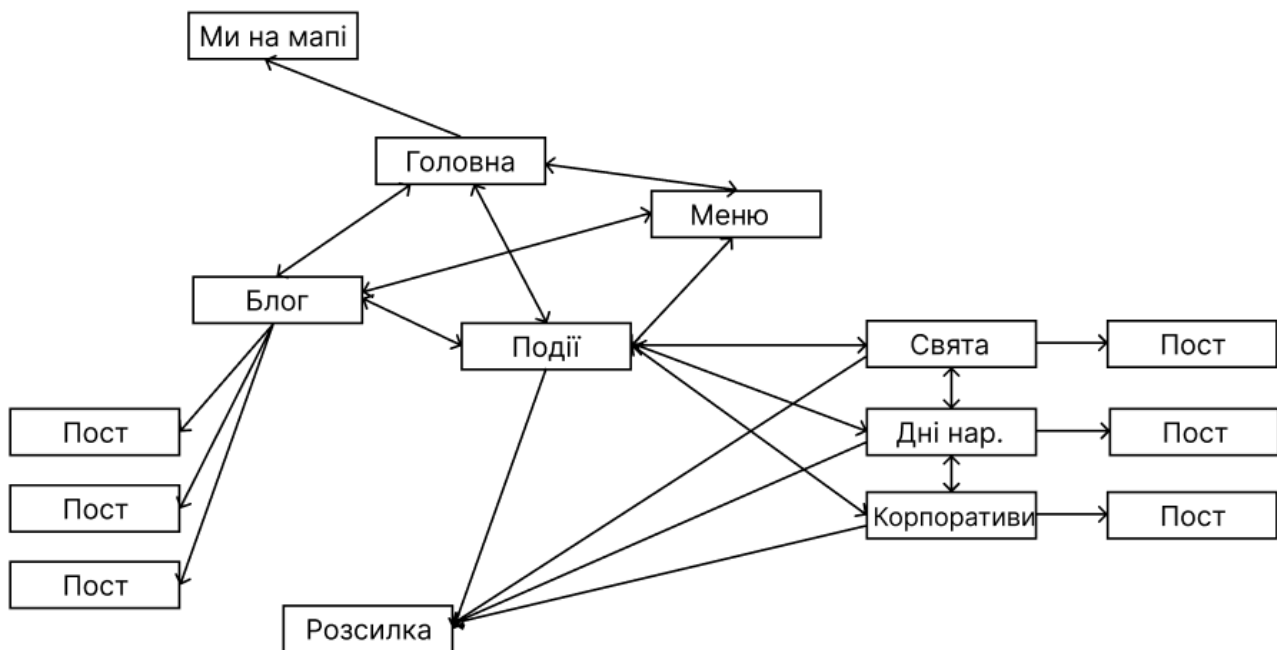


Рис 2.3. – Схема структури системи типу хмарочос для ресторану

На схемі структури я показав зв'язки сторінок між собою.

Структура сайту складається з чотирьох основних сторінок:

- Головна;
- Блог;
- Події;
- Меню.

Більш детальна структура сторінки Події відображена нижче (рис. 2.4.).

Саме ці сторінки відображають основний вміст сайту. Вони взаємопов'язані, з будь-якої з них можливо потрапити на іншу напряму.

За специфікою структури типу хмарочос, існують сторінки сайту до яких не можливо потрапити з будь-якого місця на сайті. Щоб отримати доступ до постів, ми маємо перейти на сторіну Блог. З інших сторінок сайту зробити цього не можливо.

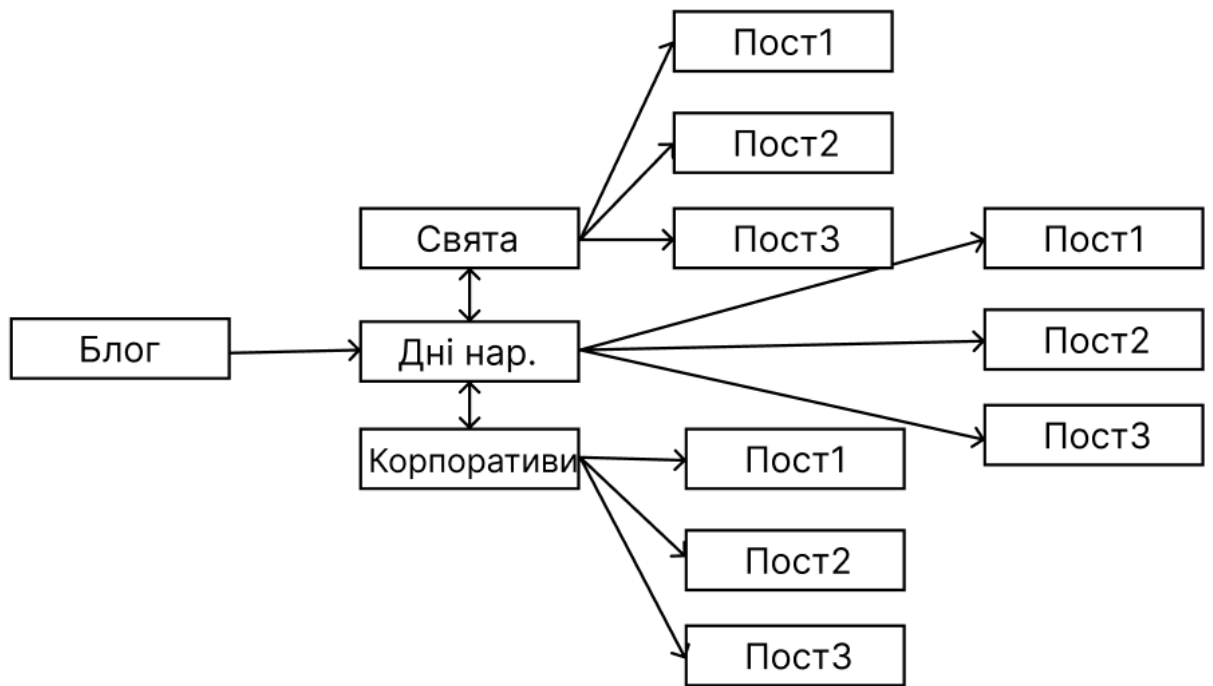


Рис 2.4. – Розгорнута схема структури сторінки Блог

Не менш важливим було обрати тип веб-орієнтованої системи за доступністю сервісів.

Так як інформаційна система була розроблена для ресторану, вміст сайту та усі сервіси є доступними для всіх користувачів та відвідувачів. Не потрібно реєструватися, щоб переглянути вміст системи. Тому дана інформаційна система є повністю відкритою.

Веб-орієнтована інформаційна система створена за допомогою бібліотеки React. А отже за компонентним підходом.

Кожен компонент системи існує самостійно та може використовуватися в будь-якому місці. Компонент – це не обов'язково один елемент, це може бути великий шматок коду зі своїми функціями та призначенням.

Зазвичай, вміст шапки чи підвалу сайту не відрізняється від того, на якій сторінці ви знаходитесь зараз. Саме тому доцільно створити з них компоненти, що будуть легчі в використанні та будуть економити більше рядків коду. А це, в свою чергу, пришвидшує роботу всієї системи.

З метою використання коду повторно, було розроблено ряд компонентів для зручного користування та масштабування системи (рис. 2.5.).

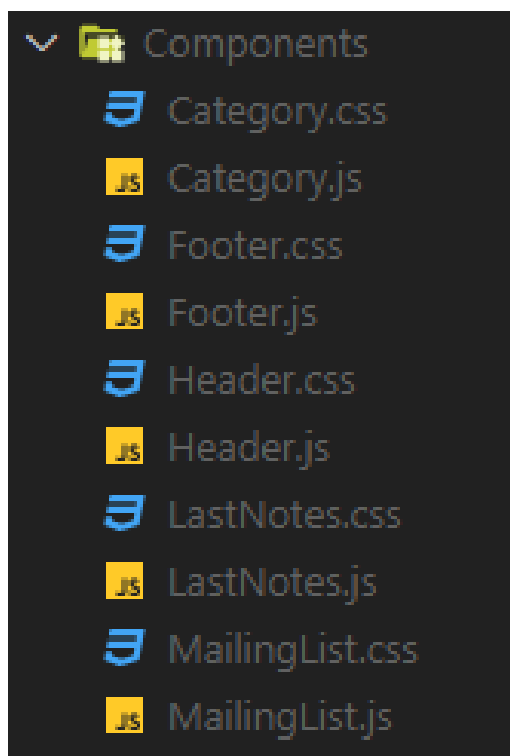


Рис 2.5. – Структура компонентів сайту

Загальна ж структура каталогу (рис. 2.6.) має три основних розділи:

- node_modules;
- public;
- src.

Кожен з даних розділів відповідає за свої конкретні функції.

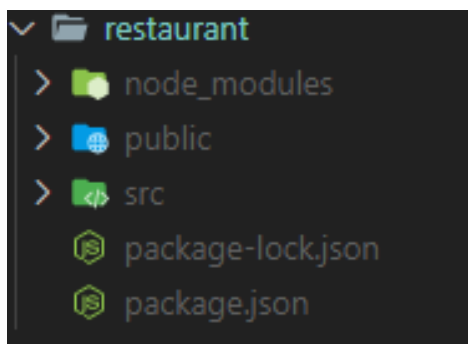


Рис 2.6. – Основні розділи структури системи

Node_modules – розділ в якому зберігається налаштування npm. За допомогою них створюється сиситема.

Розділ public (рис. 2.7.) містить html документ та фавікон до нашої інформаційної системи.

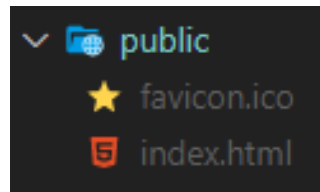


Рис 2.7. – Вміст розділу public

Розділ src (рис. 2.8.) містить в собі всі елементи та компоненти інформаційної системи, ілюстрації та фотографії, використані в системі, сторінки даного сайту.

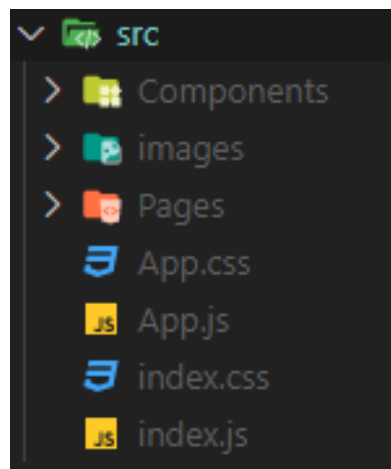


Рис 2.8. – Вміст розділу src

Також розділ src містить підрозділ pages, де зберігаються всі файли створених сторінок (рис. 2.9.).

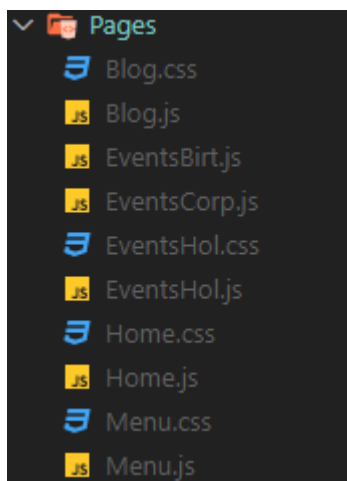


Рис 2.9. – Вміст розділу Pages

2.5. Обґрунтування та організація вхідних та вихідних даних програми

Так як за результатами даної кваліфікаційної роботи створюється повністю відкритий сайт ресторану, ніякі клієнтські дані не потрібні для відвідування та перегляду вмісту даної інформаційної системи. Тому єдиними вхідними даними системи є електронна адреса (рис. 2.10.), що надається користувачем, при бажанні, для підписки на електронну розсилку від ресторану про події та акції.

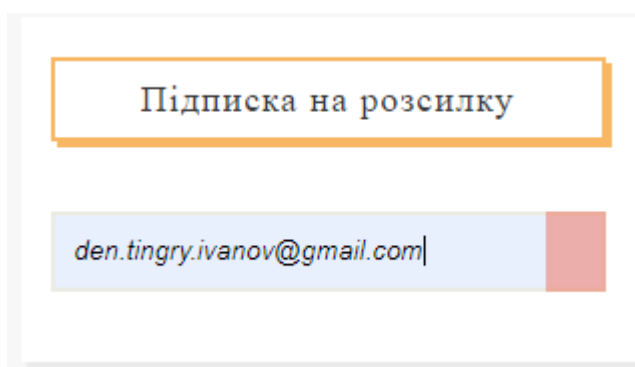


Рис 2.10. – Приклад вхідної інформації електронної адреси

Вихідними даними інформаційної системи слугують файли cookies.

Файл cookie – це текстовий файл, що зберігається якусь інформацію вашим браузером якийс проміжок часу.

Всі ці файли поділяються на сеансові та постійні. В нашій інформаційній системі файли cookies є постійними. Вони запам'ятовують електронну адресу (рис. 2.11.), та при наступному відвіданні користувача будуть надавати все записані в cookies результати для зручності користувачів.

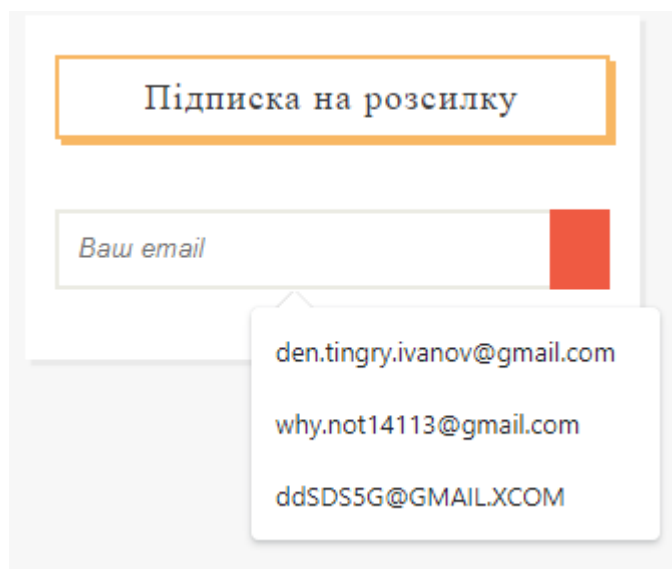


Рис 2.11. – Вихідні дані з файлів cookies

2.6. Опис розробленої системи

Для роботи з інформаційною системою, користувачу достатньо ввести в будь-якому наявному в нього браузері посилання на сайт. Після чого у вкладці браузеру відкриється інформаційна система повністю готова до використання (рис.2.12.).

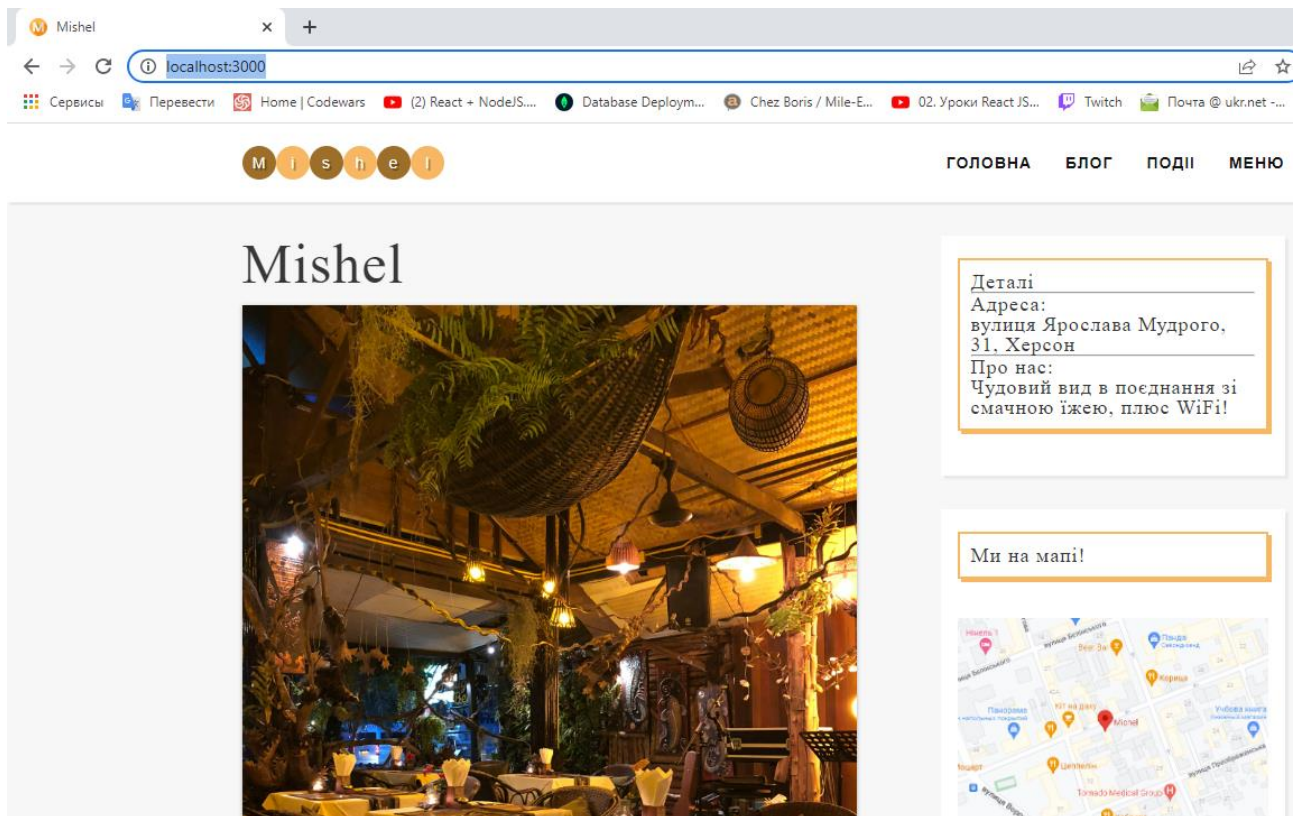


Рис 2.12. – Приклад стрічки запиту

2.6.1. Використані технічні засоби

При розробці веб-орієнтованої інформаційної системи більшість сучасних розробників рекомендують користуватися ЕОМ з такими мінімальними характеристиками:

- цп: Pentium 4;
- відеоадаптер: 3D адаптер nVidia, Intel, AMD;
- відеопам'ять: 256 МБ;
- накопичувач: 100 Гб;
- оперативна пам'ять: 2 ГБ.

При розробці даної інформаційної системи мною було використано персональну ЕОМ з наступними технічними характеристиками та пристроями:

- монітор;

- клавіатура
- комп'ютерна миша
- оперативна пам'ять 16 ГБ
- накопичувач на 480 ГБ
- відеоадаптер: nVidia;
- відеопам'ять: 6 ГБ;

2.6.2. Використані програмні засоби

При розробці даної інформаційної системи були використанні наступні програмні засоби:

- VSCode;
- Figma;
- Chrome Web Tools.

В якості середовища для написання коду я обрав один з найпопулярніших редакторів вихідного коду Visual Studio Code (рис. 2.13.).

VSCode – це безкоштовна платформа для зручного редагування та написання коду. На сьогодні він підходить для користування на найпопулярніших операційних системах: Windows, Linux, macOS.

Для більш комфортного використання платформи VSCode було встановлено наступні розширення:

- Live Server;
- VSCode Great Icons;
- Auto Close Tab;
- Vim;

Плагін Live Server дозволяє бачити зміни наживо при внесенні змін до файлів типу js, css та html (рис. 2.14.).

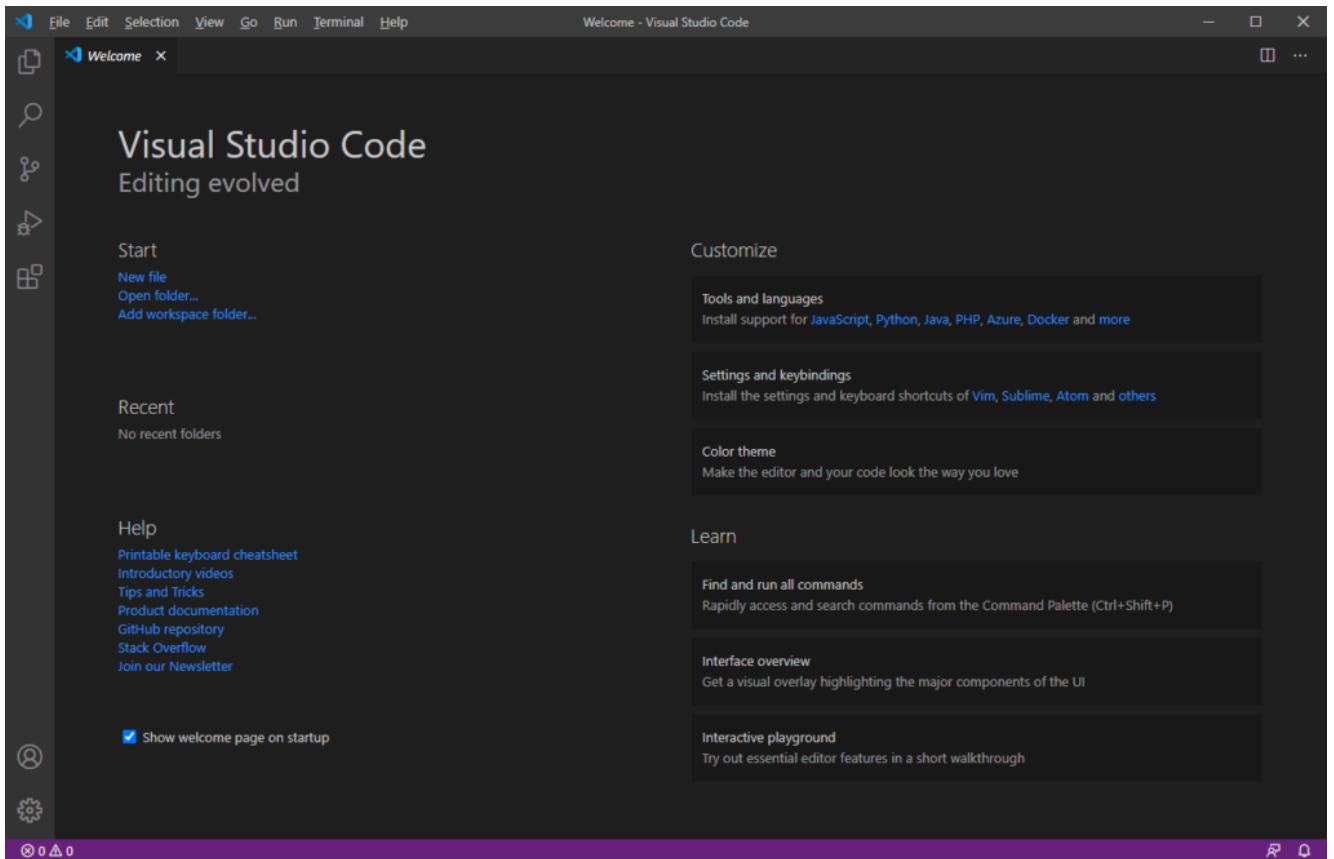


Рис 2.13. – Головна сторінка середовища VSCode

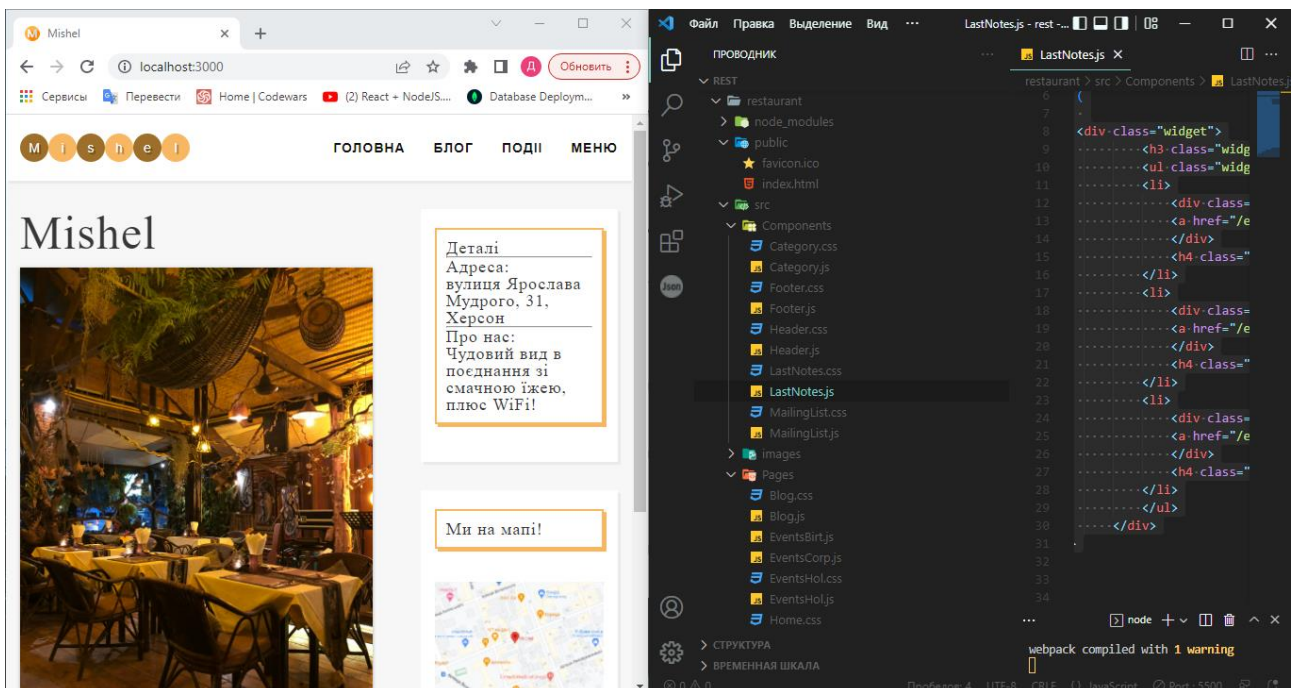


Рис 2.14. – Реалізація розширення Live Server

При створенні дизайну сайту я використовував онлайн-сервіс для розробки інтерфейсів – Figma.

Даний редактор зручний та зрозумілий в користуванні, при онлайн користуванні підтримується великою частиною сучасних браузерів. А при необхідності може бути завантажений додатком на вашу операційну систему, якщо слід працювати офлайн.

За допомогою даного графічного редактору (рис. 2.15.) можна створювати веб-застосунки, окремі елементи інтерфейсу користувачів, в моєму випадку шапка та підвал сайту.

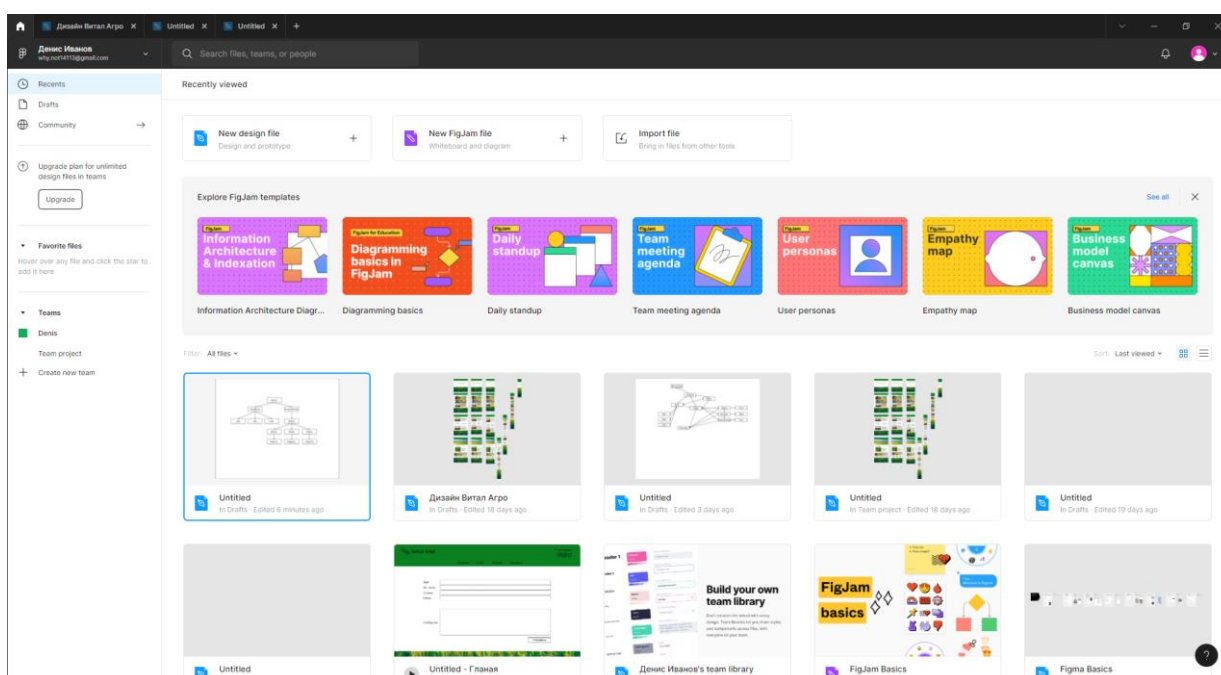


Рис 2.15. – Головна сторінка офлайн додатку Figma

Chrome Web Tools – це безкоштовні, вбудовані в браузер Chrome веб-інструменти. Вони слугують для створення та відладки сайтів.

Також, однією з їх переваг слугує можливість перегляду вихідного коду та відладки всього фронтенду (рис. 2.16.) .

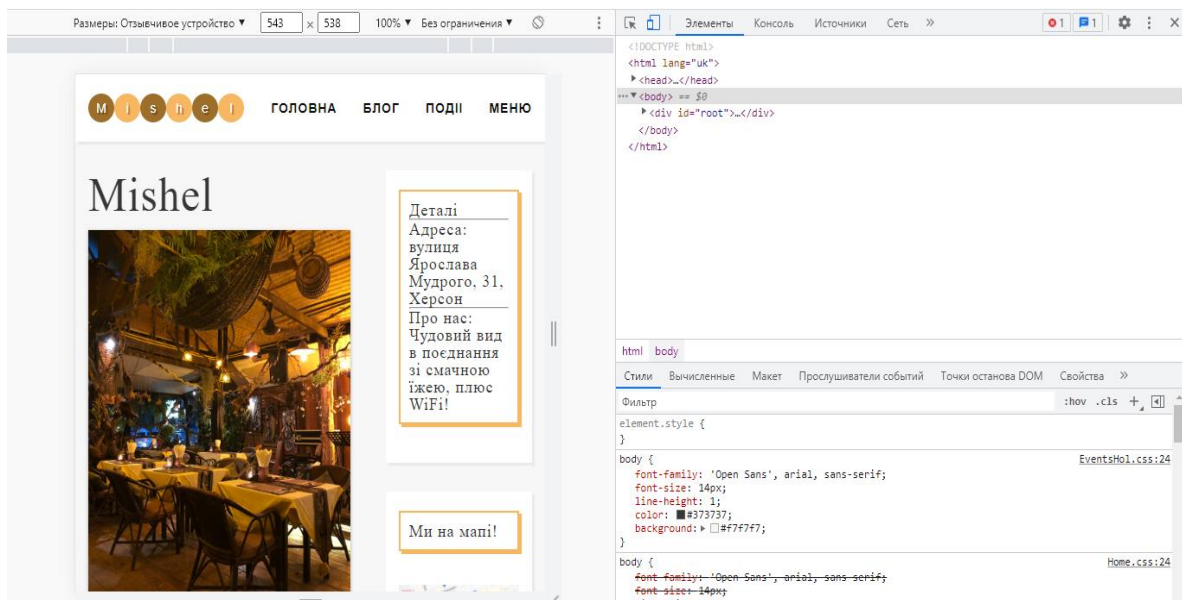


Рис 2.16. – Приклад вбудованих інструментів в браузер

2.6.3. Виклик та завантаження програми

Розроблена веб-орієнтована інформаційна система завантажується у вашому браузері в будь-якій операційній системі при введенні в адресний рядок браузера URL сайту.

Додаток гарантовано запускається на більшості браузерів останніх версій, що офіційно підтримуються розробниками.

2.6.4. Опис інтерфейсу користувача

Так як дана інформаційна система є веб-орієнтованою. Слід просто знати URL запит та ввести його в браузері в стрічку запити. Після чого ми одразу опиняємося на головній сторінці (рис. 2.17) нашого сайту.

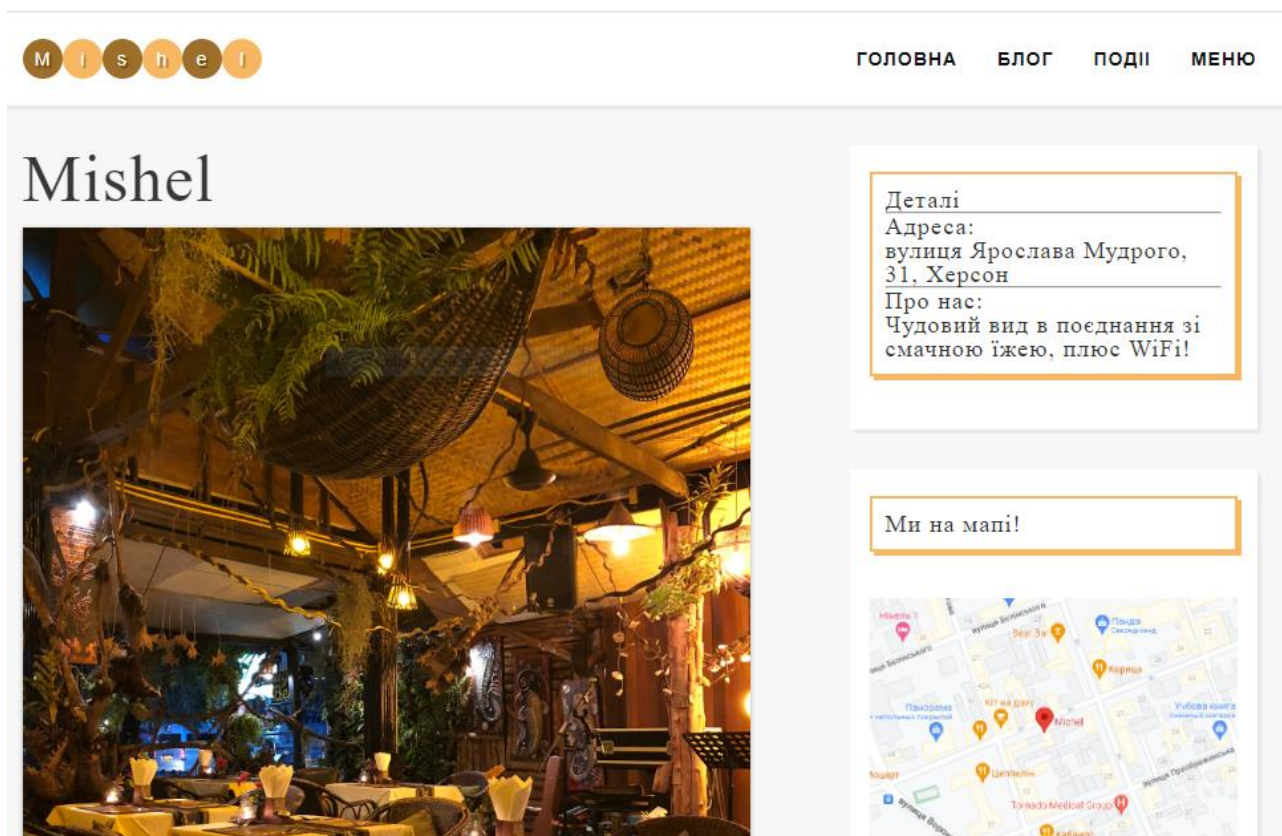


Рис 2.17. – Головна сторінка сайту

Для навігації по сторінках користувач баче навігацію в шапці сайту. Далі все просто та інтуїтивно.

При переході на сторінку “Блог” користувач баче останні події з життя ресторану та має посилання (рис. 2.18) для більш детального перегляду подій в соціальних мережах закладу.



День Вишиванки

Приєднуйтеся до святкування Дня Вишиванки разом з рестораном Mishel. На вас чекаю безліч розваг, жива музика, найсмачніші українські традиційні страви, чудова атмосфера.

ПРОДОВЖИТИ ЧИТАННЯ

Рис 2.18. – Посилання на статтю

При перегляді сторінки “Події” корситувач баче активності та свята (рис. 2.19), що пройшли нещодавно в закладі.

M i s h e l

ГОЛОВНА БЛОГ ПОДІЇ МЕНЮ

Категорії

- Свята (2)
- Дні народження (3)
- Корпоративи (1)

Останні записи

- День поцілунку
- Ювілей
- Корпоратив

СВЯТА

День молоді

28 червня ресторан "Mishel" запрошує всіх студентів відвідати заклад в честь Дня Молоді.

Рис 2.19. – Вміст сторінки Події

Для перегляду тільки тих розділів, що цікаві користувачю, створено бокову навігаційну панель (рис. 2.20) з категоріями свят та кількістю постів в кожній з них.

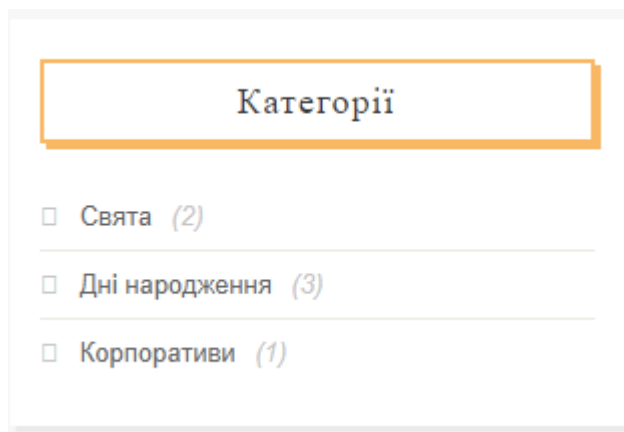


Рис 2.20. – Навігаційна панель для подій

Для того, щоб користувач міг швидко побачити найновіші події в будь-якій зі сторінок подій, було створено елемент за останніми записами (рис. 2.21).

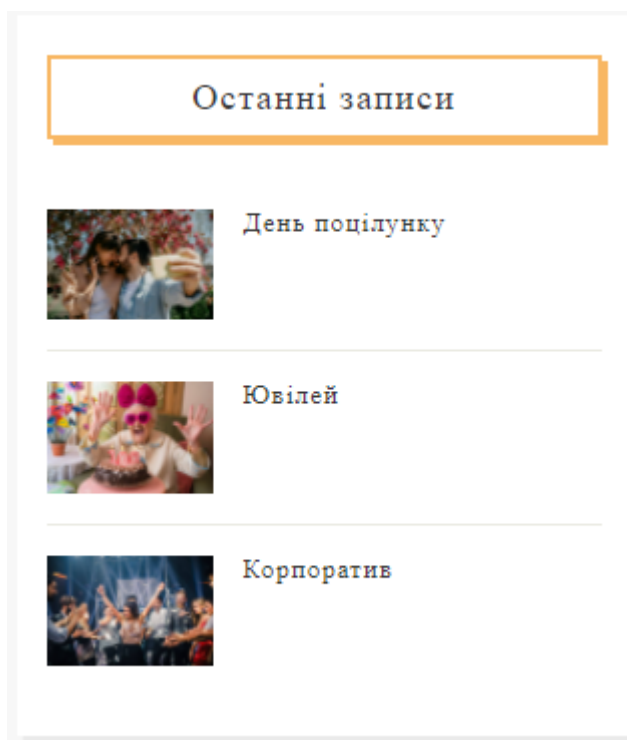


Рис 2.21. – Елемент для перегляду найактуальніших подій

Також при переході на сторінку Меню, користувач баче меню закладу з основними стравами, закусками та напоями (рис. 2.22).

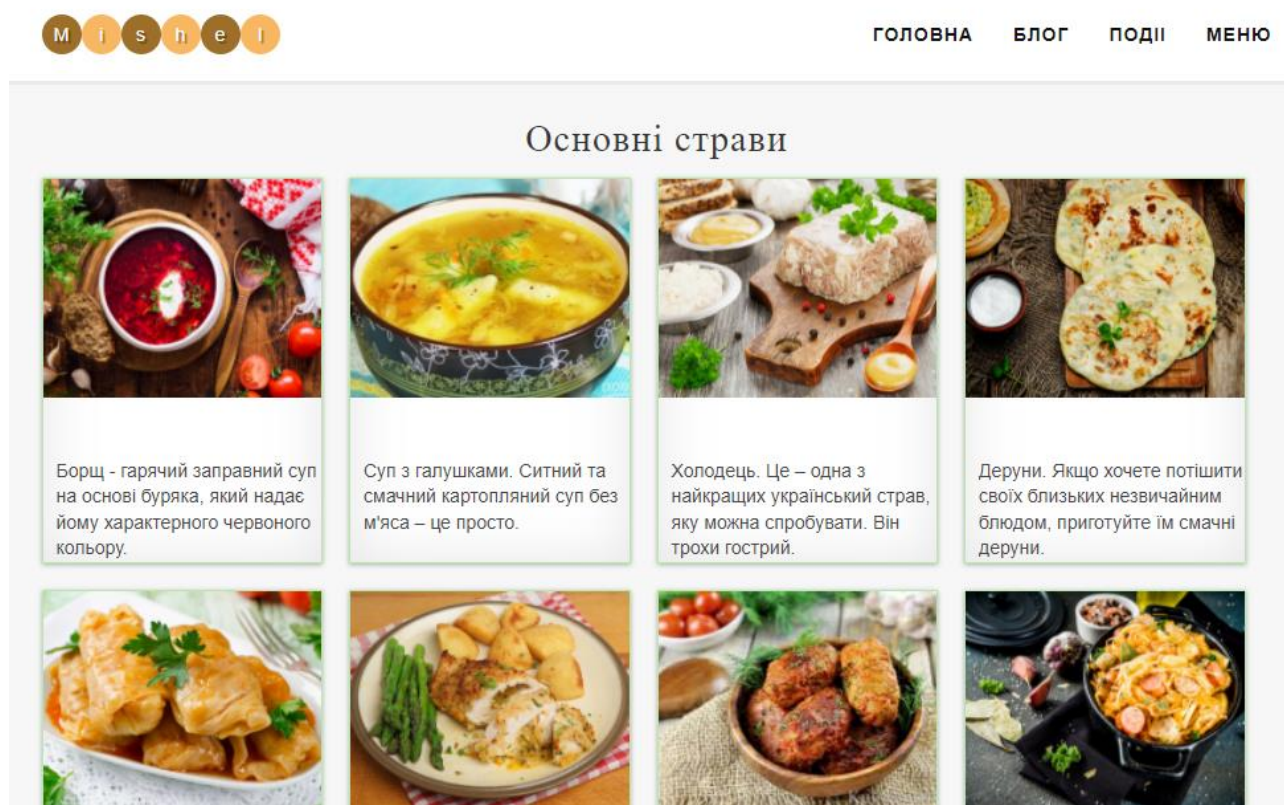


Рис 2.22. – Вміст сторінки Меню

Для зручного користування інформаційною системою, кожна сторінка сайту має адаптивність. Тобто може бути адаптована до пристроїв з меншим розширенням екранів.

Сторінка з меню ресторану адаптується до пристроїв з розширення в 400 пікселів для зручності користування на смартфонах (рис. 2.23).

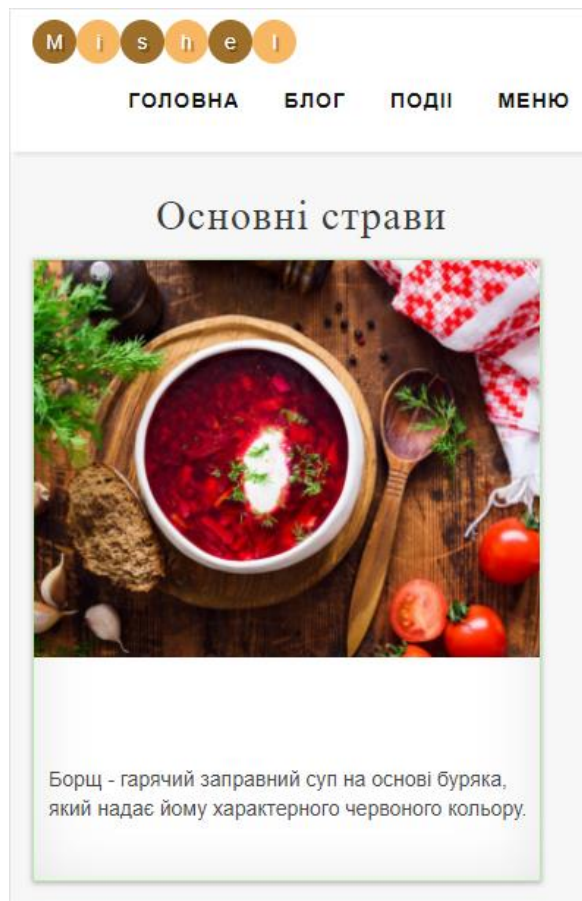


Рис 2.23. – Приклад адаптивності вмісту сторінок

За таким же принципом адаптовані всі інші сторінки нашоїго сайту (рис. 2.24-26).



Рис 2.24. – Адаптована Головна сторінка

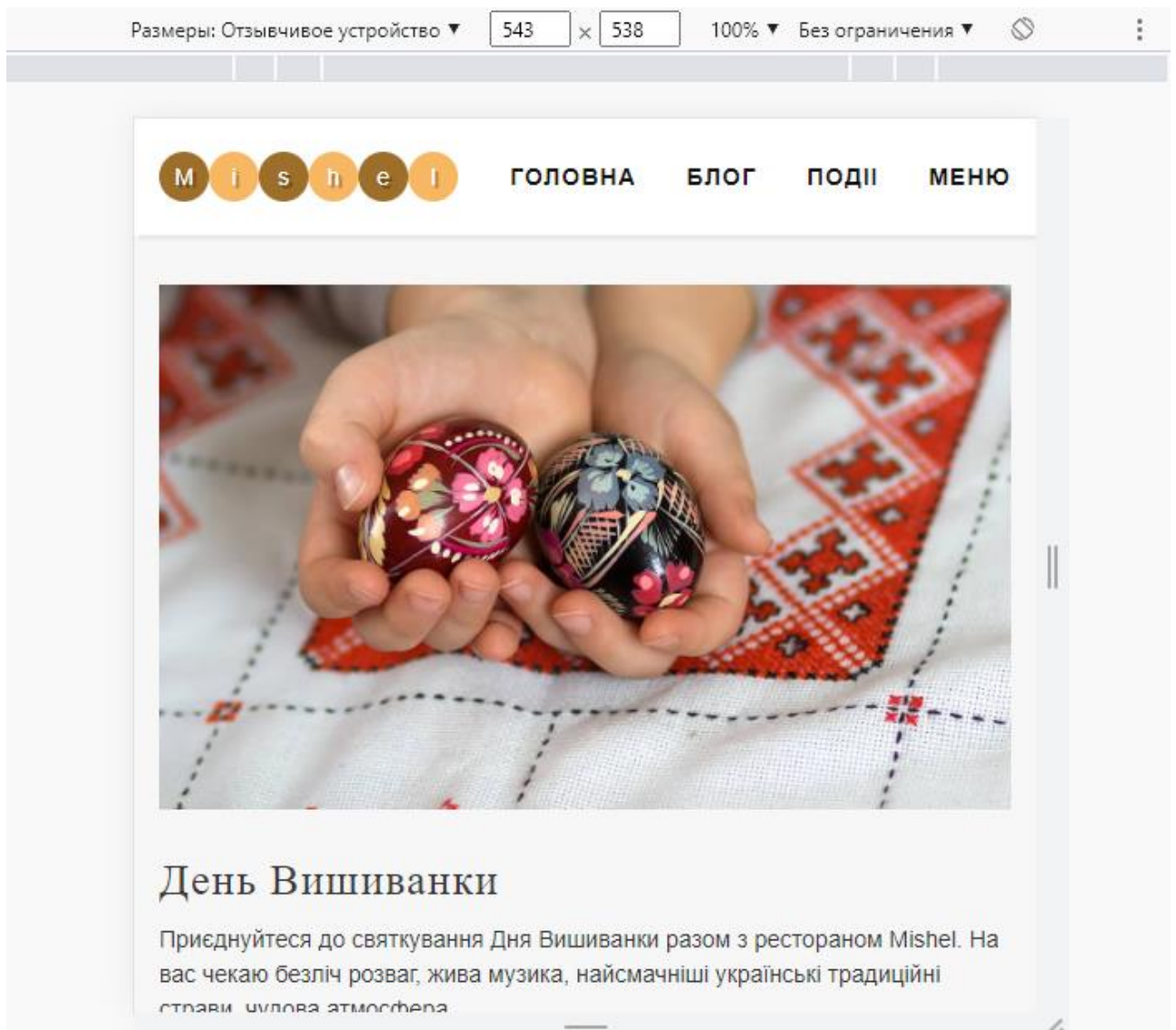


Рис 2.25. – Адаптована сторінка блогу

Дані сторінки автоматично підганяються під розміри екранів пристроїв користувачів.

Тому наш заклад не втрачає клієнтів, в яких наявний тільки ПК.

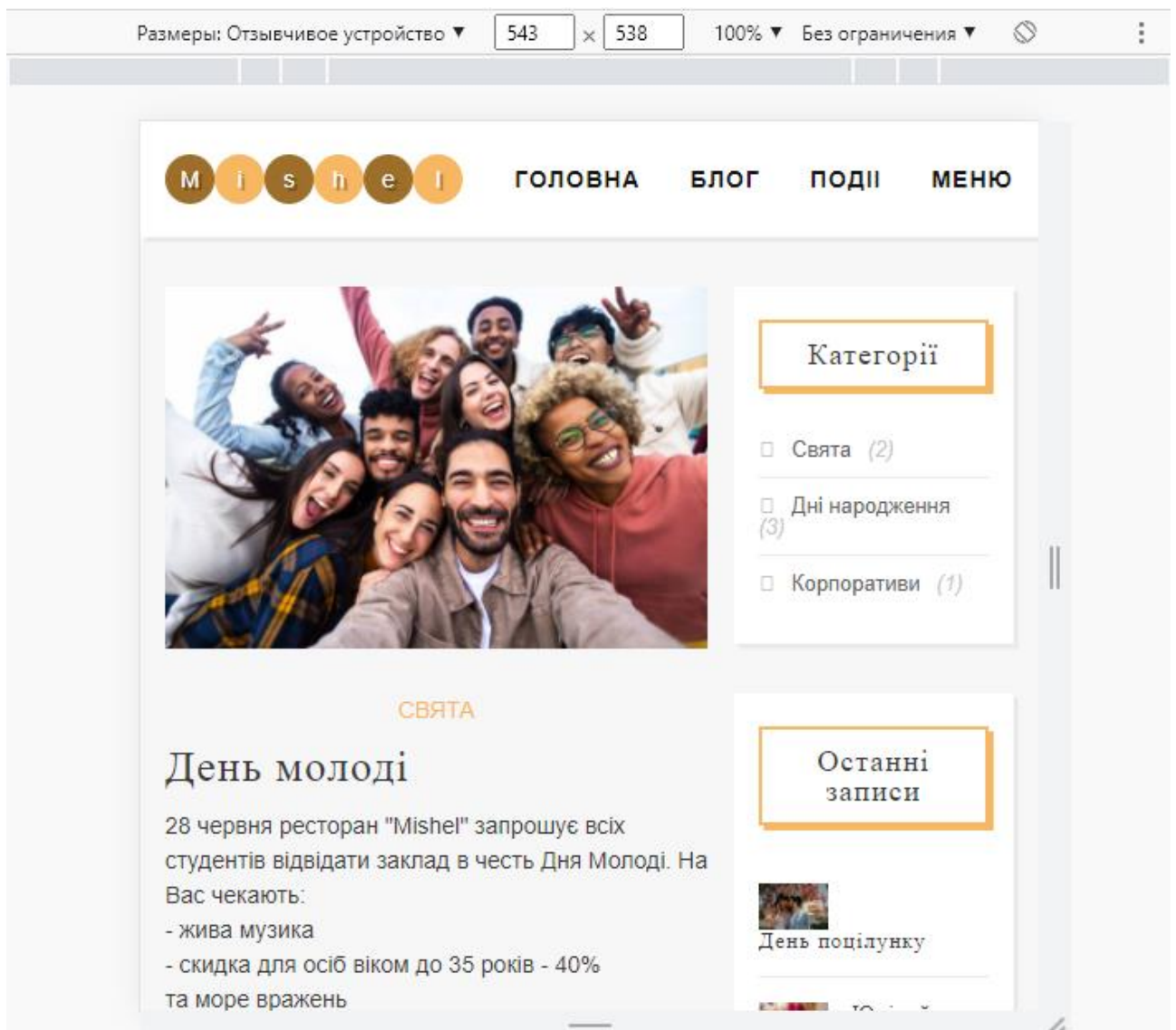


Рис 2.26. – Адаптована сторінка подій та активностей

РОЗДІЛ 3

ЕКОНОМІЧНИЙ РОЗДІЛ

3.1 Розрахунок трудомісткості та вартості розробки програмного продукту

Вихідні дані:

1. передбачуване число операторів програми – 1516;
2. коефіцієнт кореляції програми в ході її розробки – 0,1;
3. коефіцієнт складності програми – 1,2;
4. годинна заробітна плата програміста– 128 грн/год;

Середня заробітна плата програміста степеню Junior Frontend Developer розраховувалась за даними сайту «Rabota.UA» [16].

За даними початку 2022 року середня заробітна плата фахівця становить 800\$.

Довідковий курс Національного банку України з долару США 28 січня становить 28.2 грн. За 1 долар, а отже 22560 грн.

Враховуючи це, при графіку в 40 робочих годин на тиждень та при 176 годинах на місяць, заробітна плата за одну робочу годину становитиме 128 гривень.

5. коефіцієнт збільшення витрат праці в наслідок недостатнього опису задачі – 1.2;
6. коефіцієнт кваліфікації програміста, обумовлений стажем праці з даної спеціальності – 1.3;
7. вартість машино-години ЕОМ – 16 грн/год;

При розробці даного проекту для покращення технічних можливостей програміста для розробки програмного продукту хорошим рішенням було купити достатньо потужний ПК. Вартість якого складала 30000 грн. Тому вартість машино години ЕОМ враховує ці витрати.

8. кількість операторів програми – 1.

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:

$$t = t_o + t_u + t_a + t_n + t_{omл} + t_{\partial} , \text{ людино-годин, } (3.1)$$

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

t_u - витрати праці на дослідження алгоритму рішення задачі;

t_a - витрати праці на розробку блок-схеми алгоритму;

t_n - витрати праці на програмування по готовій блок-схемі;

$t_{omл}$ - витрати праці на налагодження програми на ЕОМ;

t_{∂} - 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 (1516);

C - коефіцієнт складності програми (1.2);

p - коефіцієнт кореляції програми в ході її розробки (0.1).

Підставивши відомі дані в формулу маємо:

$$Q = 1.2 \cdot 1516 \cdot (1 + 0.1) = 2001.12$$

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

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

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

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

$$t_u = (2001.12 \cdot 1.2) / (75 \cdot 1.3) = 24.6 \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі:

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

де: Q – умовне число операторів програми;

k – коефіцієнт кваліфікації програміста.

Підставивши дані значення в формулу (3.4), отримаємо:

$$t_a = 2001.12 / (20 \cdot 1.3) = 76.9 \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

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

Підставивши дані значення в формулу (3.5), отримаємо:

$$t_n = 2001.12 / (25 \cdot 1.3) = 61.5 \text{ людино-годин}$$

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:

$$t_{oml} = 2001.12 / (5 \cdot 1.3) = 307.8 \text{ людино-годин.}$$

– 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:

$$t_{oml}^k = 1.5 \cdot 307.8 = 461.7 \text{ людино-годин.}$$

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:

$$t_{\partial p} = 2001.12 / (16 \cdot 1.3) = 96.2 \text{ людино-годин.}$$

$$t_{\partial o} = 0.75 \cdot 96.2 = 72.15 \text{ людино-годин.}$$

$$t_{\partial} = 96.2 + 72.15 = 168.35 \text{ людино-годин.}$$

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

$$t = 50 + 24.6 + 76.9 + 61.5 + 461.7 + 168.35 = 843 \text{ людино-години.}$$

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:

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

Заробітна плата виконавців визначається за формулою:

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

де: t - загальна трудомісткість, людино-годин;

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

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

$$З_{зп} = 843 \cdot 128 = 107\,904 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$З_{мв} = t_{отл} \cdot C_{мч}, \text{ грн, (3.13)}$$

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:

$$З_{мв} = 461.7 \cdot 13 + 30000 = 36002.1 \text{ грн.}$$

На основі отриманих даних витрати на створення програмного продукту наступні:

$$K_{ПО} = 107\,904 + 36002.1 = 143\,906.1 \text{ грн.}$$

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:

$$T = 843 / (1 \cdot 176) \approx 4.78 \text{ міс.}$$

Висновок: на розробку програмного продукту, що призначений для ознайомлення користувачів з інформаційною системою, необхідно 843 людино-години. З цих даних маємо, що ймовірна тривалість розробки продукту складатиме 4.78 місяці при стандартному робочому тижні та робочому місяці. Вартість даного програмного продукту становитиме 143 906 грн. В дану суму входить заробітна плата фахівця та затрати на ЕОМ, а також його покупку.

ВИСНОВОК

Під час виконання даної кваліфікаційної роботи був розроблений веб-орієнтована система, що викорситовувала сучасні підходи та інструменти для її створення.

Інформаційна система призначена для інформування користувачів з новинками та подіями закладу. Також вона відіграє велику роль в маркетингу та приваблює більше користувачів й відвідувачів до ресторану.

Дана веб-система має практичну цінність для користувачів. Через те, що економить їх час. Їм не потрібно відвідувати заклад для перегляду меню та активностей.

За рахунок впровадження сучасних інструментів дана інформаційна система має можливість покращення.

Під час виконання даної кваліфікаційної роботи були поставлені та виконані наступні задачі:

- аналіз предметної області;
- знайдено підстави для розробки даної інформаційної системи;
- обрано раціональні та сучасні структуру та технології створення;
- написано програмний код веб-системи;
- розроблено рекомендації щодо використання системи.

Також у ході виконання кваліфікаційної роботи було визначено трудомісткість розробленого програмного продукту (843 людино-години), проведено розрахунок вартості розробки програмного продукту (143000 грн) та розраховано час на його створення (4.78 місяці).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. React Documentation «Основи React». [Електронний ресурс]. Режим доступу: <https://reactjs.org/docs/getting-started.html>
2. React Router «Основи роутингу в React». [Електронний ресурс]. Режим доступу: <https://www.geeksforgeeks.org/reactjs-router>
3. React Components «Компоненти React». [Електронний ресурс]. Режим доступу: https://www.w3schools.com/react/react_components.asp
4. The 5 basics of websites building «Основи створення веб-сайтів». [Електронний ресурс]. Режим доступу: <https://learn.g2.com/website-building>
5. Sagar Ganatra. React Router Quick Start Guide: Routing in React applications made easy, 2018. – 158p.
6. Css documentation «Основи Css». [Електронний ресурс]. Режим доступу: <https://developer.mozilla.org/eng/docs/Web/CSS>
7. Jon Duckett. HTML and CSS: Design and Build Websites - Packt Publishing, 2012. - 512p.
8. Ben 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.
9. Julie C. Meloni. HTML, CSS, and JavaScript All in One: Covering HTML5, CSS3, and ES6, Sams Teach Yourself 7rd Edition, 2019. - 1500p.
10. JS documentation «Основи JS». [Електронний ресурс]. Режим доступу: <https://developer.mozilla.org/ru/docs/Web/JavaScript>
11. Luis Atencio. The Joy of JavaScript Packt Publishing, 2021. - 360p.
12. Noel Rappin. Modern Front-end development for Rails, second edition, 2021. – 375p.
13. Jessica Minnick. Responsive Web Design with HTML 5 & CSS (MindTap Course List) 9th Edition. 2020. – 640p.
14. Chong Lip Phang. Mastering Front-End Web Development: 14 Books in 1. Introducing 200+ Extensions. An Advanced Guide. 2020. – 457p.

15. Cross-browser Layout techniques «Кросс-браузерні методи макетування». [Електронний ресурс]. Режим доступу: <https://www.oreilly.com/library/view/beginning-css3>
16. Rabota.ua «Почук інформації про заробітні плати». [Електронний ресурс]. Режим доступу: <https://rabota.ua/ua>
17. Website Style Guide – How to Create a Web Design Style Guide «Як обрати стиль сайту». [Електронний ресурс]. Режим доступу: <https://uxhacks.com/website-style-guide/>
18. VSCode Documentation «VSCode документація». [Електронний ресурс]. Режим доступу: <https://code.visualstudio.com/docs>
19. VSCode Extensions «VSCode розширення». [Електронний ресурс]. Режим доступу: <https://blog.bitsrc.io/top-10-visual-studio-code-extensions-for-web-developers-5bd6a76bdf5f>
20. Figma documentation «VSCode документація». [Електронний ресурс]. Режим доступу: <https://www.figma.com/community>

КОД ПРОГРАМИ

Package-lock.json

```
{
  "name": "restaurant",
  "version": "0.1.0",
  "lockfileVersion": 2,
  "requires": true,
  "packages": {
    "": {
      "name": "restaurant",
      "version": "0.1.0",
      "dependencies": {
        "@testing-library/jest-dom": "^5.16.4",
        "@testing-library/react": "^13.2.0",
        "@testing-library/user-event": "^13.5.0",
        "react": "^18.1.0",
        "react-dom": "^18.1.0",
        "react-router-dom": "^6.3.0",
        "react-scripts": "5.0.1",
        "web-vitals": "^2.1.4"
      }
    }
  }
}
```

Index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
```

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

App.js

```
import Footer from './Components/Footer';
import Header from './Components/Header';
import './App.css';
```

```
function App() {
  return (
    <>
```



```

    <Header />
    <Footer />
  </>
)
}

export default App;

```

Index.html

```

<!DOCTYPE html>
<html lang="uk">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="./favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <title>Mishel</title>
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>

```

Category.js

```

import React, { Component } from 'react';
import './Components/Category.css'

export default class Category extends Component {
  render () {
    return (
      <>
        <div class="widget">
          <h3 class="widget-title">Категорії</h3>
          <ul class="widget-category-list">
            <li><a href="/events/hol">Свята</a> (2)</li>
            <li><a href="/events/birt">Дні народження</a> (3)</li>
            <li><a href="/events/corp">Корпоративи</a> (1)</li>
          </ul>
        </div>
      </>
    )
  }
}

```

LastNotes.js

```

import React, { Component } from 'react';
import './Components/LastNotes.css'

export default class LastNotes extends Component {
  render () {
    return (
      <div class="widget">
        <h3 class="widget-title">Останні записи</h3>
        <ul class="widget-posts-list">
          <li>
            <div class="post-image-small">
              <a href="/events/hol"><img src={require('./images/post2.jpg')} alt='День
молоді' /></a>
            </div>
            <h4 class="widget-post-title">День поцілунку</h4>
          </li>
          <li>
            <div class="post-image-small">
              <a href="/events/birt"><img src={require('./images/100.jpg')} alt='День
народження' /></a>
            </div>
            <h4 class="widget-post-title">Ювілей</h4>
          </li>
          <li>
            <div class="post-image-small">
              <a href="/events/birt"><img src={require('./images/corp1.jpg')}
alt='Корпоратив' /></a>
            </div>
            <h4 class="widget-post-title">Корпоратив </h4>
          </li>
        </ul>
      </div>
    )
  }
}

```

MailingList.js

```

import React, { Component } from 'react';
import './Components/MailingList.css'

export default class MailingList extends Component {
  render () {
    return (
      <div class="widget">
        <h3 class="widget-title">Підписка на розсилку</h3>
        <form action="https://www.ukr.net/" id="subscribe" target="_blank">
          <input type="email" name="email" placeholder="Ваш email" required />
        </form>
      </div>
    )
  }
}

```

```

        <button type="submit">
          <a href='https://www.ukr.net/'></a>
        </button>
      </form>
    </div>
  </>
)
}
}

```

Footer.js

```

import React, { Component } from 'react';
import './Components/Footer.css'
import { BrowserRouter, Routes, Route } from "react-router-dom";

export default class Footer extends Component {
  render () {
    return (
      <>
        <footer>
          <div className="container">
            <div className="footer-col">
              <span>Ресторан Mishel<br></br><p></p><br></br> © 2022</span>
            </div>

            <div className="footer-col">
              <span>Наша адреса:<br></br>вулиця Ярослава Мудрого, 31,
Херсон,<br></br> Херсонська область, Україна, 73009</span>
              <div className="social-bar-wrap">
                </div>
            </div>

            <div className="footer-col">
              <span>
                Бронювання номерів:
                <p></p>
                <br><p>+380666472113</p></span>
            </div>
          </div>
        </footer>
      </>
    )
  }
}

```

Header.js

```

import React, { Component } from 'react';
import './Components/Header.css'

```

```

import { BrowserRouter, Routes, Route } from "react-router-dom";

import Menu from '../Pages/Menu';
import Blog from '../Pages/Blog';
import Home from '../Pages/Home';
import EventsHol from '../Pages/EventsHol';
import EventsBirt from '../Pages/EventsBirt';
import EventsCorp from '../Pages/EventsCorp';

export default class Header extends Component {
  render () {
    return (
      <>
        <header>
          <nav className="container">
            <a className="logo" href="/">
              <span>M</span>
              <span>i</span>
              <span>s</span>
              <span>h</span>
              <span>e</span>
              <span>l</span>
            </a>
            <div className="nav-toggle"><span></span></div>

            <ul id="menu">

              <li><a href="/">Головна</a></li>
              <li><a href="/blog">Блог</a></li>
              <li><a href="/events/hol">Події</a></li>
              <li><a href="/menu">Меню</a></li>
            </ul>
          </nav>
        </header>
        <BrowserRouter>
          <Routes>
            <Route path="/" exact element={ <Home /> } />
            <Route path="/blog" exact element={ <Blog /> } />
            <Route path="/menu" exact element={ <Menu /> } />
            <Route path="/events/birt" exact element={ <EventsBirt /> } />
            <Route path="/events/hol" exact element={ <EventsHol /> } />
            <Route path="/events/corp" exact element={ <EventsCorp /> } />
          </Routes>
        </BrowserRouter>
      </>
    )
  }
}

```

Blog.js

```

import React, { Component } from 'react';
import './Pages/Blog.css'
import vish from './images/dayofV.jpg'
import kherson from './images/dayOfKhe.jpg';

export default class Blog extends Component {
  render () {
    return (
      <>
        <div className="container">
          <div className="posts-list-blog">
            <article id="post-1" className="post">
              <div className="post-image">
                <a href="https://twitter.com/" target="_blank"><img src={ vish } /></a>
              </div>
              <div className="post-content">
                <h2 className="post-title">День Вишиванки</h2>
                <p>Приєднуйтеся до святкування Дня Вишиванки разом з рестораном
                Mishel. На вас чекаю безліч розваг, жива музика, найсмачніші українські традиційні страви,
                чудова атмосфера. </p>
                <div className="post-footer">
                  <a className="more-link" href="https://twitter.com/"
target="_blank">Продовжити читання</a>
                </div>
              </div>
            </article>
            <article id="post-2" className="post">
              <div className="post-image">
                <a href="https://twitter.com/" target="_blank"><img src={ kherson }
/></a></div>
              <div className="post-content">
                <h2 className="post-title">День Херсону</h2>
                <p>Свято Дня міста Херсон відзначають у третю декаду вересня. Рік
                заснування Херсона – 1778. Традиційні святкові заходи, як правило, розраховані на
                найширшу аудиторію.</p>
                <div className="post-footer">
                  <a className="more-link" href="https://twitter.com/"
target="_blank">Продовжити читання</a>
                </div>
              </div>
            </article>
          </div>
        </div>
      </>
    )
  }
}

```

EventBirt.js

```

import React, { Component } from 'react';

```

```

import '../Pages/EventsHol.css';
import MailingList from '../Components/MailingList';
import LastNotes from '../Components/LastNotes';
import Category from '../Components/Category';

export default class EventsBirt extends Component {
  render () {
    return (
      <div class="container">
        <div class="posts-list">
          <article id="post-1" class="post">
            <div class="post-image"><img src={require('../images/100.jpg')} alt='День
народження' /></div>
            <div class="post-content">
              <div class="category"><a href="">Дні народження</a></div>
              <h2 class="post-title">Ювілей у 100 років</h2>
              <p>4 вересня наш заклад вітає Галину Івановну з її ювілеєм.</p>
              <div class="post-footer">
                <a class="more-link" href="https://twitter.com/" target="_blank">Продовжити
читання</a>
              </div>
            </div>
          </article>
          <article id="post-2" class="post">
            <div class="post-image"><img src={require('../images/13.jpg')} alt='День
народження' /></div>
            <div class="post-content">
              <div class="category"><a href="">Дні народження</a></div>
              <h2 class="post-title">День народження Дениса</h2>
              <p>Вітаємо з Днем Народження Дениса, та сподіваємося, що його
святкування буде неперевершеним в нашому закладі.</p>
              <div class="post-footer">
                <a class="more-link" href="https://twitter.com/" target="_blank">Продовжити
читання</a>
              </div>
            </div>
          </article>
          <article id="post-3" class="post">
            <div class="post-image"><img src={require('../images/22.jpg')} alt='День
народження' /></div>
            <div class="post-content">
              <div class="category"><a href="">Дні народження</a></div>
              <h2 class="post-title">День народження Сергія</h2>
              <p>Зі святом Сергію. Від всього персоналу закладу бажаємо Вам успіхів, і
приємного проведення часу в ресторані Mishel.</p>
              <div class="post-footer">
                <a class="more-link" href="https://twitter.com/" target="_blank">Продовжити
читання</a>
              </div>
            </div>
          </article>
        </div>
      </div>
    );
  }
}

```

```

        </div>
      </div>
    </article>
  </div>
  <aside>
    <Category />
    <LastNotes />
    <MailingList />
  </aside>
</div>
</>
)
}
}

```

EventCorp.js

```

import React, { Component } from 'react';
import MailingList from '../Components/MailingList';
import LastNotes from '../Components/LastNotes';
import Category from '../Components/Category';
import '../Pages/EventsHol.css';
export default class EventsCorp extends Component {
  render () {
    return (
      <>
        <div class="container">
          <div class="posts-list">
            <article id="post-1" class="post">
              <div class="post-image">
                <img src={require('../images/corp1.jpg')} alt='Корпоратив' />
                <img src={require('../images/corp2.jpg')} alt='Корпоратив' />
              </div>
              <div class="post-content">
                <div class="category"><a href="">Корпоративи</a></div>
                <h2 class="post-title">Корпоратив компанії КОРС</h2>
                <p>В честь 10-річчя заснування компанії "КОРС" 18 грудня відбувся
корпоратив з живою музикою...</p>
                <div class="post-footer">
                  <a class="more-link" href="https://twitter.com/"
target="_blank">Продовжити читання</a>
                </div>
              </div>
            </article>
          </div>
          <aside>
            <Category />
            <LastNotes />
            <MailingList />
          </aside>
        </div>
      </>
    );
  }
}

```

```

    </div>
  </>
)
}
}

```

EventHol.js

```

import React, { Component } from 'react';
import { BrowserRouter, Routes, Route } from "react-router-dom";

import '../Pages/EventsHol.css'
import MailingList from '../Components/MailingList';
import LastNotes from '../Components/LastNotes';
import Category from '../Components/Category';

export default class EventsHol extends Component {
  render () {
    return (
      <div class="container">
        <div class="posts-list">
          <article id="post-1" class="post">
            <div class="post-image"><img src={require('../images/post1.jpg')} alt='День
молоді' /></div>
            <div class="post-content">
              <div class="category"><a href="">Свята</a></div>
              <h2 class="post-title">День молоді</h2>
              <p>28 червня ресторан "Mishel" запрошує всіх студентів відвідати
заклад в честь Дня Молоді. На Вас чекають: <br></br> - жива музика
<br></br> - скидка для осіб віком до 35 років - 40%<br></br>та море
вражень</p>
              <div class="post-footer">
                <a class="more-link" href="https://twitter.com/"
target="_blank">Продовжити читання</a>
              </div>
            </div>
          </article>
          <article id="post-2" class="post">
            <div class="post-image"><img src={require('../images/post2.jpg')} alt='День
поцілунку' /></div>
            <div class="post-content">
              <div class="category"><a href="">Свята</a></div>
              <h2 class="post-title">Всесвітній день поцілунку</h2>
              <p>Ресторан "Mishel" запрошує всіх охочів в День поцілунку відвідати
нас. Для пар - скидка 25% на напої.</p>
              <div class="post-footer">
                <a class="more-link" href="https://twitter.com/"
target="_blank">Продовжити читання</a>
              </div>
            </div>
          </article>
        </div>
      </div>
    )
  }
}

```



```

        </div>
      </article>
    </div>
  <aside>
    <Category />
    <LastNotes />
    <MailingList />
  </aside>
</div>
</>
)
}
}

```

Home.js

```

import React, { Component } from 'react';

import './Pages/Home.css';
import map from './images/map.png';
import picHome from './images/imagHome.jpg';

export default class Home extends Component {
  render () {
    return (
      <>
        <div class="container">
          <div class="posts-list-home">
            <article class="post-home">
              <div class="post-content-home">
                <h1 class="post-title-home">Mishel</h1>
              </div>
              <div class="post-image-home"><img src={ picHome } /></div>

            </article>
          </div>
          <aside>
            <div class="widget-home">
              <h4 class="widget-title-home">Деталі<br/><hr/>
              Адреса: <br/>
              вулиця Ярослава Мудрого, 31, Херсон<hr/>
              Про нас: <br/>
              Чудовий вид в поєднання зі смачною їжею, плюс WiFi!

            </h4>
          </div>
          <div class="widget-home">
            <h3 class="widget-title-home">Ми на мапі!</h3>
            <ul class="widget-posts-list-home">
              <li>
                <div >

```

```

        <a
href="https://www.google.com.ua/maps/place/Michel/@46.6359679,32.6065033,17z/data=!4m5!3
m4!1s0x40c41ab7e56c687d:0xf6e6ed28c5c91095!8m2!3d46.6358906!4d32.6081555?hl=ru"
target="_blank"><img src={ map } /></a>
    </div>
  </li>
</ul>
</div>
</aside>
</div>
</>
)
}
}

```

Menu.js

```

import React, { Component } from 'react';
import '../Pages/Menu.css'
<img src={require('../images/syrniki.png')} alt="" />

export default class Menu extends Component {
  render () {
    return (
      <div class="container">
        <div class="category">
          <h1>Основні страви</h1>
        </div>
        <ul class="products clearfix">
          <li class="product-wrapper">
            <a href="" class="product">
              <div class="product-photo">
                <img src={require('../images/borsh2.png')} alt='Борщ' />
              </div>
              <p className='product_text'>Борщ - гарячий заправний суп на основі
                буряка, який надає йому характерного червоного кольору.</p>
            </a>
          </li>
          <li class="product-wrapper">
            <a href="" class="product">
              <div class="product-photo">
                <img src={require('../images/galush.png')} alt='Суп з галушками' />
              </div>
              <p className='product_text'>Суп з галушками. Ситний та смачний
                картопляний суп без м'яса – це просто.</p>
            </a>
          </li>
          <li class="product-wrapper">
            <a href="" class="product">
              <div class="product-photo">

```

```

        <img src={require('../images/holoddec.png')} alt='Холодець' />
    </div>
    <p className='product_text'>Холодець. Це – одна з найкращих український
страв, яку можна спробувати. Він трохи гострий.
    </p>
    </a>
</li>
<li class="product-wrapper">
    <a href="" class="product">
        <div class="product-photo">
            <img src={require('../images/derynu.png')} alt='Деруни' />
        </div>
        <p className='product_text'>Деруни. Якщо хочете потішити своїх
близьких незвичайним блюдом, приготуйте їм смачні деруни.
        </p>
    </a>
</li>
<li class="product-wrapper">
    <a href="" class="product">
        <div class="product-photo">
            <img src={require('../images/golybci.png')} alt='Голубці' />
        </div>
        <p className='product_text'>Голубці. Традиційно голубці готують з
використанням фаршу з яловичини, змішаного з рисом.
        </p>
    </a>
</li>
<li class="product-wrapper">
    <a href="" class="product">
        <div class="product-photo">
            <img src={require('../images/kotleta.png')} alt='Котлета по-Київські' />
        </div>
        <p className='product_text'>Котлети по-київськи. Дуже популярна страва
в усьому світі.</p>
    </a>
</li>
<li class="product-wrapper">
    <a href="" class="product">
        <div class="product-photo">
            <img src={require('../images/greshaniki.png')} alt='Гречаники' />
        </div>
        <p className='product_text'>Гречаники. Страва з'явилася в українській
кухні порівняно недавно, причому в різних і несподіваних варіаціях.</p>
    </a>
</li>
<li class="product-wrapper">
    <a href="" class="product">
        <div class="product-photo">
            <img src={require('../images/bigos.png')} alt='Бігос' />
        </div>
        <p className='product_text'>Бігос. Бігос – насправді перлина польської
кухні, хоча його національність оскаржують і литовці.</p>
    </a>
</li>

```

```

    </a>
  </li>

  <div class="category">
    <h1>Закуски</h1>
  </div>
  <li class="product-wrapper">
    <a href="" class="product">
      <div class="product-photo">
        <img src={require('../images/salo.png')} alt='Солене сало' />
      </div>
      <p className='product_text'>Солене сало. Рецепти засолювання сала, до
яких ви будете повертатися знову і знову</p>
    </a>
  </li>
  <li class="product-wrapper">
    <a href="" class="product">
      <div class="product-photo">
        <img src={require('../images/kapusta.png')} alt='Капуста маринована' />
      </div>
      <p className='product_text'>Маринована капуста. Одна з
найпопулярніший закусок України</p>
    </a>
  </li>
  <li class="product-wrapper">
    <a href="" class="product">
      <div class="product-photo">
        <img src={require('../images/shponder.png')} alt='Шпондер' />
      </div>
      <p className='product_text'>Шпондер. М'ясистий свинячий підчеревок,
запечений в печі на дровах або в духовці.</p>
    </a>
  </li>
  <li class="product-wrapper">
    <a href="" class="product">
      <div class="product-photo">
        <img src={require('../images/seledka.png')} alt='Селедка по-галицьки' />
      </div>
      <p className='product_text'>Селедка по-галицьки. М'ясистий свинячий
підчеревок, запечений в печі на дровах або в духовці.</p>
    </a>
  </li>

  <div class="category">
    <h1>Напої</h1>
  </div>
  <li class="product-wrapper">
    <a href="" class="product">
      <div class="product-photo">
        <img src={require('../images/americano.png')} alt='Американо' />
      </div>

```

```

        <p className='product_text'>Американо. Американо – один із найбільш
популярних кавових напоїв у світі.</p>
    </a>
</li>
<li class="product-wrapper">
    <a href="" class="product">
        <div class="product-photo">
            <img src={require('./images/fresh.png')} alt='Апельсиновий фреш' />
        </div>
        <p className='product_text'>Апельсиновий фреш. Свіжий апельсиновий сік
з макіттю.</p>
    </a>
</li>
<li class="product-wrapper">
    <a href="" class="product">
        <div class="product-photo">
            <img src={require('./images/pivo.png')} alt='Крафтове пиво' />
        </div>
        <p className='product_text'>Крафтове пиво. Чудовий унікальний смак ви
не забудете надовго.</p>
    </a>
</li>
<li class="product-wrapper">
    <a href="" class="product">
        <div class="product-photo">
            <img src={require('./images/hotchokolate.png')} alt='Гарячий шоколад' />
        </div>
        <p className='product_text'>Гарячий шоколад. Приготовлений із
розтопленого шоколаду, іноді називають питним шоколадом</p>
    </a>
</li>
</ul>
</div>
</>
)
}
}

```

Решту файлів можна переглянути в прикріпленому до роботи носії на якому буде архів з повною роботою.

ВІДГУК

керівника економічного розділу
на кваліфікаційну роботу бакалавра

на тему:

«Розробка програмного додатку «Система електронного голосування»»

студента групи 122-17-1 Ліс'їх Артема Ігоровича

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

Л. В. Касьяненко

Перелік файлів на диску

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

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