

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

_____ проф. Приходько С.Б.

«___» _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня вищої освіти «бакалавр»

на тему: Розробка програмного забезпечення вебсайту для
ресторану «International»

Виконав: студент 4151 групи



_____ Петров В.В.

(підпис, ПІБ)

Керівник роботи:

_____ Викладач, к.т.н.

(посада, науковий ступень вчене звання)

_____ Пухалевич А.В.

(підпис, ПІБ)

Миколаїв – 2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»
Гарант освітньої програми
_____ доц. Макарова Л.М.
(підпис)
«___» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «бакалавр»

Студенту Петров Віталій Віталійович
(Прізвище, ім'я, по батькові)

1. Тема роботи: Розробка програмного забезпечення вебсайту для _____ ресторану
«International»

Керівник роботи Пухалевич Андрій Володимирович
(Прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора №256-уч від «11» травня 2022 р.

2. Термін подання роботи: 10.06.2022 р.

3. Вихідні дані по роботі: _____

4. Перелік питань, що належать до розробки (найменування розділів)

Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Опис предметної галузі та постановка задачі; Проект програмного забезпечення (ескізний, технічний та робочий); Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів 1. Титульний слайд, 2. Мета кваліфікаційної роботи, 3. Аналіз предметної області та постановка задачі на розробку ПЗ вебсайту, 4. Діаграма варіантів використання, 5. Діаграма діяльності, 6. Діаграма «Сутність-Зв'язок», 7. Діаграма класів, 8. Діаграма послідовності та логічна модель, 9. Обґрунтування вибору засобів програмування, 10. Результат розробки програмного забезпечення вебсайту для ресторану «International», 11. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

7. Дата видачі завдання 11.05.2022 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу Вступ	24.03.2022	ПП*
2.	Опис та аналіз предметної галузі	28.03.2022	ПП*
3.	Постановка задачі	31.03.2022	ПП*
4.	Ескізний проєкт програмного забезпечення	27.04.2022	
5.	Технічний проєкт програмного забезпечення	06.05.2022	
6.	Робочий проєкт програмного забезпечення	13.05.2022	
7.	Підготовка розділу Результати розробки програмного забезпечення	17.05.2022	
8.	Підготовка розділу з охорони праці	18.05.2022	ПП*
9.	Підготовка розділу Висновки	19.05.2022	
10.	Оформлення списку використаних джерел та додатків	20.05.2022	
11.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	10.06.2022	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку)
12.	Підготовка презентації та доповіді	12.06.2022	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	14.06.2022	
14.	Подання на кафедру ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді	24.06.2022	

* - за результатами переддипломної практики (ПП), яка була з 21.03.2022 до 08.05.2022 р.

Студент

(підпис)

Петров В.В.
(прізвище та ініціали)

Керівник роботи

(підпис)

Пухалевич А.В.
(прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота присвячена розробці програмного забезпечення вебсайту для ресторану «International». Дана робота пов'язана з автоматизацією та клієнтоорієнтованістю системи створення замовлення, можливістю створення облікового запису, зберігання інформації про замовлення та елементів у меню, ознайомлення зі списком меню та пошук додаткової інформації про ресторан.

У кваліфікаційній роботі проведено аналіз предметної області та методів створення програмного забезпечення вебсайту для ресторану «International». Виконано постановку задачі на розробку програмного забезпечення вебсайту для ресторану «International».

Також кваліфікаційна робота містить:

- технічне завдання;
- ескізний проєкт;
- робочий проєкт;
- методика та програма випробувань, а також результати виконання випробувань;
- інструкція користувача, системи вимог;
- розділ з охорони праці;
- опис програмного забезпечення.

Кваліфікаційну роботу виконано на 123 сторінках друкованого тексту, містить 66 рисунків, 21 таблиць, 5 додатків, та список використаних джерел із 27 найменування.

ABSTRACT

The qualifying work is devoted to the development of website software for the restaurant "International". This work is related to the automation and customer orientation of the order creation system, the ability to create an account, store information about the order and menu items, read the menu list and search for additional information about the restaurant.

The qualification work analyzes the subject area and methods of creating website software for the restaurant "International". The task of developing a website software for the International restaurant has been completed.

Also the qualifying work contains:

- technical task;
- sketch project;
- working project;
- test methodology and program, as well as test results;
- user manual, system requirements;
- section on labor protection;
- software description.

Qualification work was performed on 123 pages of printed text, contains 66 figures, 21 tables, 5 appendices, and a list of used sources from 27 names.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ВИМОГ ТА ІСНУЮЧИХ ВЕБСАЙТІВ ДЛЯ РЕСТОРАНІВ. ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ	8
1.1 Аналіз вимог до вебсайту на базі браузеру для ресторану «International».....	8
1.2 Аналіз існуючих аналогів та обґрунтування доцільності розробки програмного забезпечення вебсайту.....	9
1.3 Постановка задачі на розробку вебсайту для ресторану «International».....	14
2 ПРОЄКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ ДЛЯ РЕСТОРАНУ «INTERNATIONAL».....	16
2.1 Ескізний проєкт програмного забезпечення вебсайту для ресторану «International».....	16
2.1.1 Побудова моделі варіантів використання вебсайту «International»	16
2.1.2 Конкретизація варіантів використання вебсайту «International»...	18
2.1.3 Моделювання сценаріїв основних варіантів використання вебсайту «International» за допомогою діаграми діяльності.....	22
2.1.4 Інформаційна модель даних вебсайту.....	23
2.1.5 Проєкт користувацького інтерфейсу вебсайту для ресторану «International».....	24
2.2 Технічний проєкт вебсайту на базі браузеру для ресторану «International».....	32
2.2.1 Статична модель вебсайту «International».....	32
2.2.2 Пояснення класів з діаграми класів вебсайту «International».....	33

2.2.3 Динамічна модель вебсайту «International» у вигляді діаграми послідовності.....	34
2.2.4 Логічна модель даних вебсайту.....	35
2.3 Робочий проєкт вебсайту «International».....	37
2.3.1 Обґрунтування вибору засобів програмування.....	37
2.3.2 Фізична модель даних вебсайту «International».....	38
2.3.3 Розробка діаграми розгортання вебсайту.....	41
2.3.4 Реалізація основних класів вебсайту.....	41
2.3.5 Тестування програмного забезпечення вебсайту «International»...	42
2.3.6 Випробування вебсайту.....	49
3 РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ ДЛЯ РЕСТОРАНУ «INTERNATIONAL».....	56
4 ОХОРОНА ПРАЦІ.....	57
ВИСНОВКИ.....	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	64
ДОДАТОК А ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ ДЛЯ РЕСТОРАНУ «INTERNATIONAL».....	67
ДОДАТОК Б ПРОГРАМА ТА МЕТОДИКА ВИПРОБОВУВАНЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ ДЛЯ РЕСТОРАНУ «INTERNATIONAL».....	81
ДОДАТОК В ІНСТРУКЦІЯ КОРИСТУВАЧА ДЛЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ ДЛЯ РЕСТОРАНУ «INTERNATIONAL»....	88
ДОДАТОК Г КОД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ ДЛЯ РЕСТОРАНУ «INTERNATIONAL».....	98
ДОДАТОК Д ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ ДЛЯ РЕСТОРАНУ «INTERNATIONAL».....	119

ВСТУП

Сьогодні майже кожен заклад, у тому числі ресторани, мають свій сайт. Через сайт ресторану дуже зручно переглядати його меню, контактну інформацію, місце розташування тощо. Тому розробка програмного забезпечення вебсайту для ресторану «International» має практичну цінність.

Метою кваліфікаційної роботи є розв'язання практичної задачі програмної інженерії, що характеризується комплексністю та невизначеністю умов, а саме розробка програмного забезпечення вебсайту для ресторану «International».

Для досягнення поставленої мети потрібно виконати наступні завдання:

- провести аналіз предметної області розробки програмного забезпечення вебсайту;
- виконати постановку задачі;
- розробити ескізний проєкт програмного забезпечення вебсайту;
- розробити технічний проєкт програмного забезпечення вебсайту;
- розробити робочий проєкт програмного забезпечення вебсайту;
- виконати кодування та тестування програмного забезпечення вебсайту;
- розглянути питання з охорони праці;
- створення програмної документації.

1 АНАЛІЗ ВИМОГ ТА ІСНУЮЧИХ ВЕБСАЙТІВ ДЛЯ РЕСТОРАНІВ.

ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ВЕБСАЙТУ ДЛЯ РЕСТОРАНУ «INTERNATIONAL»

1.1 Опис предметної області. Аналіз методів створення програмного забезпечення вебсайту для ресторану «International»

Ресторан «International» є закладом харчування, який пропонує широкий вибір страв. Клієнтам не завжди зручно ходити до ресторану, щоб оглянути меню, ознайомитися із рестораном, зробити замовлення чи забронювати місце. Щоб автоматизувати ці процеси, потрібно розробити програмне забезпечення вебсайту для ресторану «International», який надаватиме повну інформацію про ресторан до прибуття клієнта. Вебсайт повинен запропонувати повне меню з інформацією про страви, та можливістю забронювати місце на певний час і дату. Також вебсайт має містити опис ресторану, галерею, новини та контакти: адресу ресторану та номер телефону ресторану.

Клієнт має змогу ознайомитися з інформацією про заклад до прибуття у ресторан. У вікні опису ресторану клієнт може ознайомитися із коротким інформаційним описом, та перейти на інші соціальні мережі ресторану. Для ознайомлення з інтер'єром клієнт може через вебсайт оглянути приміщення ресторану у вікні «Галерея». Також ознайомитися із меню, та обрати власний сет замовлення. Після закінчення формування замовлення, також можна забронювати собі місце на зручний час й дату. Головні новини також будуть на сайті, а контакти ресторану зручно розташовані у кінці вебсайту.

Ресторан повинен мати декілька сторінок:

- 1) Головна сторінка, яка містить таку інформацію: опис, посилання на соціальні мережі, меню, галерею, новини, та контакти;
- 2) Перехід на сторінку меню, де клієнт може сформувати замовлення;
- 3) Сторінка яка містить: оформлення, підтвердження замовлення, та бронювання місця на певний час й дату;
- 4) Сторінка історії замовлення клієнта;
- 5) Сторінку профілю клієнта;
- 6) Окрему сторінку вебсайту для менеджера.

1.2 Аналіз наявних аналогів та обґрунтування доцільності розробки програмного забезпечення вебсайту для ресторану «International»

1.2.1 Вебсайт «Beer-Altbier»

Сайт «Beer-Altbier» має простий, але зрозумілий інтерфейс. Цей сайт має декілька сторінок: головна сторінка, меню, опис, караоке, доставлення, та кошик для оплати [1].

Знімок екрану головної сторінки вебсайту представлено на рисунку 1.1.

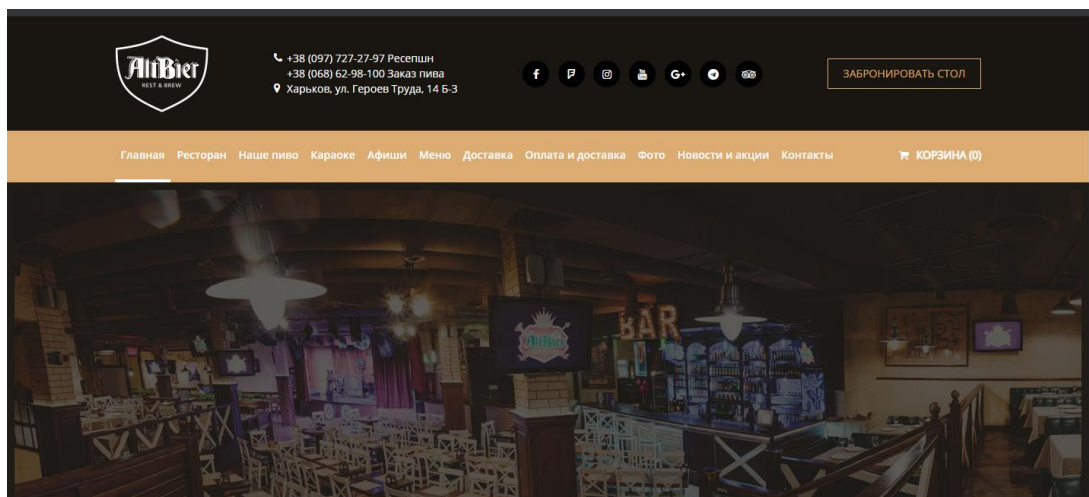


Рисунок 1.1 – Головна сторінка вебсайту «Beer-Altbier»

Знімок екрану сторінки меню вебсайту представлено на рисунку 1.2.

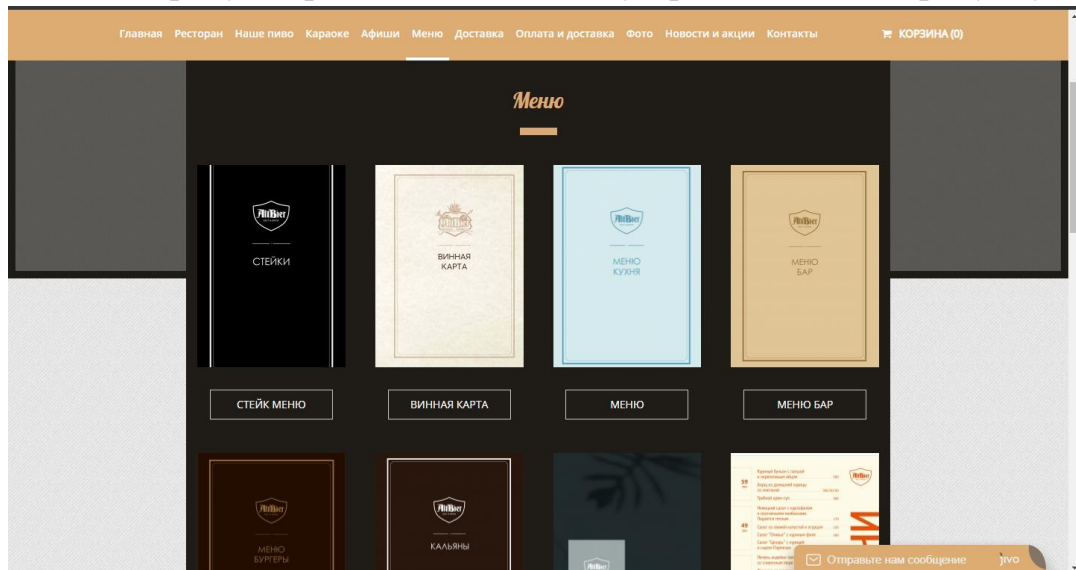


Рисунок 1.2 – Сторінка меню вебсайту «Beer-Altbiere»

Переваги сайту:

- Зручний зрозумілий інтерфейс;
- Адаптивний вебсайт на кожную платформу;
- Присутня вся інформація на сайті щодо ресторану.

Недоліки сайту:

- Багате та розбите меню;
- Багато сторінок на сайту.

1.2.2 Вебсайт «Sfood – ParaSHA!»

«Sfood – ParaSHA!» – це ресторан який пропонує швидку їжу. Вебсайт ресторану виконано на 1-2 сторінки, де на першій сторінці розташована вся інформація ресторану [2].

Знімок екрану головної сторінки вебсайту представлено на рисунку 1.3.

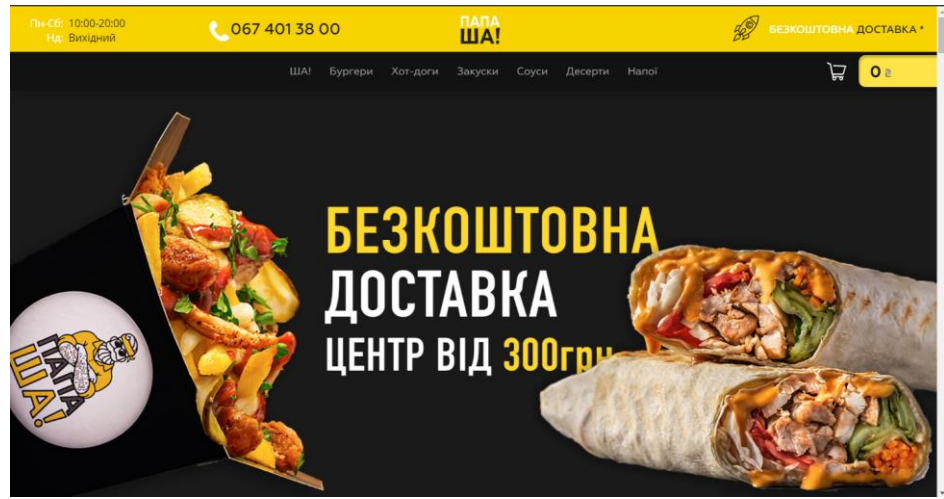


Рисунок 1.3 – Головна сторінка_вебсайту «Sfood – PapaSHA!»

Знімок екрану сторінки меню вебсайту представлено на рисунку 1.4.

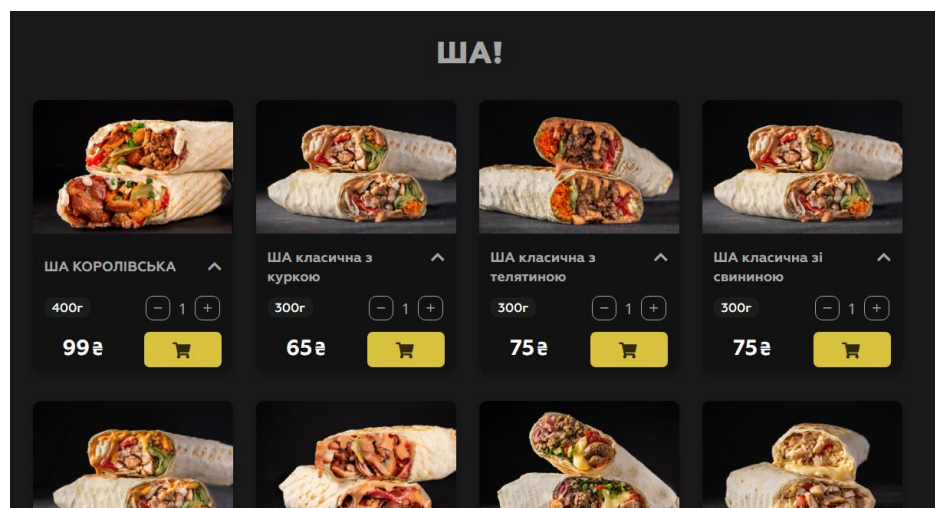


Рисунок 1.4 – Сторінка меню_вебсайту «Sfood – PapaSHA!»

Переваги сайту:

- Простий інтерфейс виконано на одну сторінку;
- Адаптивний вебсайт на кожну платформу.

Недоліки сайту:

- Незручно шукати певну їжу, приправу, блюдо, щоб розробити сет замовлення;
- Немає галереї інтер'єру ресторану, складно знайти місце.

1.2.3 Вебсайт «Zelda»

Вебсайт ресторану «Zelda» має лише одну сторінку [3]. Знімок екрану головної сторінки вебсайту представлено на рисунку 1.5.

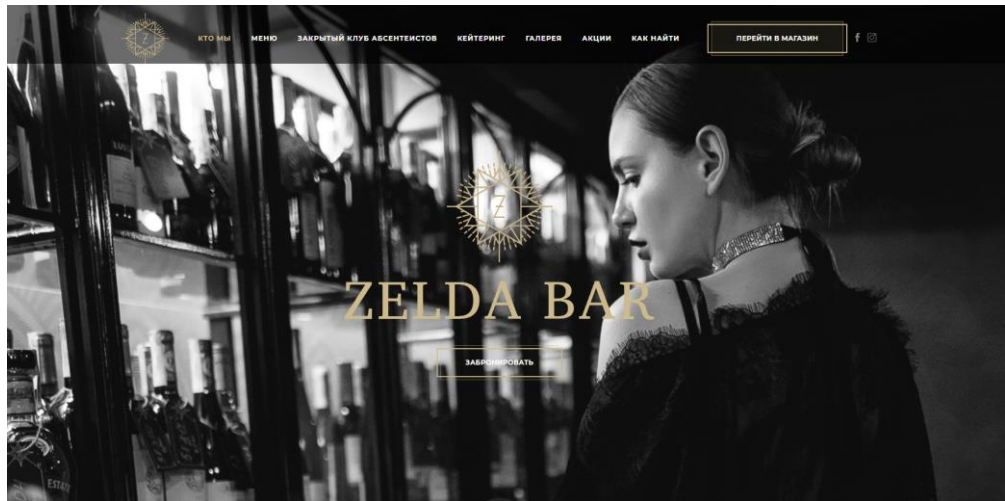


Рисунок 1.5 – Головна сторінка вебсайту «Zelda»

Знімок екрану сторінки меню вебсайту представлено на рисунку 1.6.

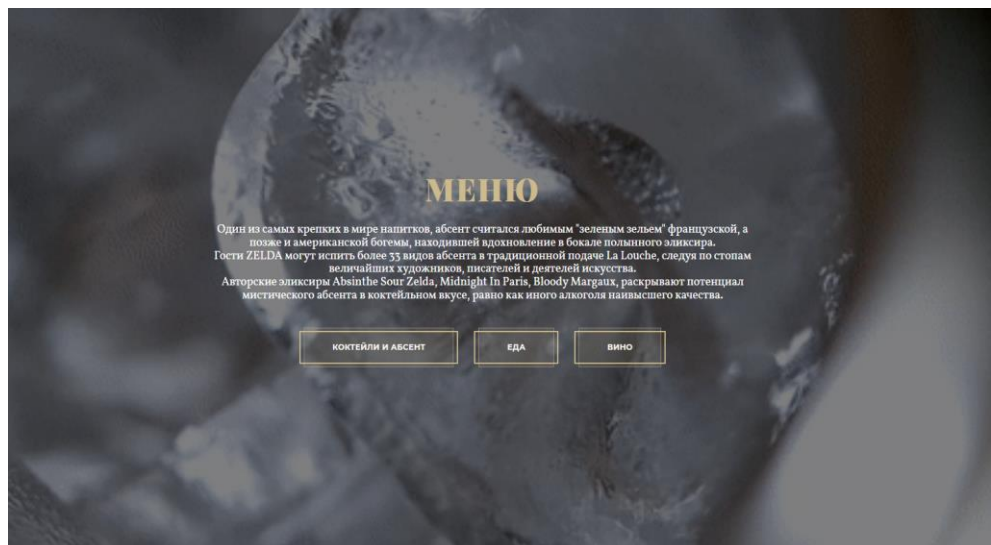


Рисунок 1.6 – Сторінка меню вебсайту «Zelda»

Переваги сайту:

- Дуже якісний дизайн інтерфейсу;
- Якісні анімації, переходи, зрозумілий, та приємний інтерфейс;
- Є можливість оглянути галерею сайту;
- Коротка та потрібна інформація.

Недоліки сайту:

- Щоб ознайомитися із меню, сайт пропонує відкрити PDF файл;
- Меню навігаційної панелі не слідує за користувачем, якщо перейти вниз сайту, тобто немає можливості повернутися уверх, чи перемикнути на іншу вкладку, потрібно переходити вручну знизу угору.

1.2.4 Обґрунтування доцільності розробки вебсайту

Аналіз аналогів вебсайту показав, що хоча вони мають практичну та виконавчу складову, але вони мають певні недоліки:

- 1) не має можливості переглянути інтер'єр закладу;
- 2) не зручне меню, яке не надає достатньої інформації або взагалі її не існує;
- 3) тяжко знайти певну страву, напій чи інший елемент з меню.

Виходячи з вищеповисаних недоліків, постає питання про розробку програмного забезпечення вебсайту для ресторану «International», який матиме наступні функції:

- Перехід на певні пункти сайту, через посилання у меню панелі навігації;
- Перегляд слайдів у галереї;
- Перехід з сайту на соціальні мережі;
- Додати елемент з меню у замовлення;
- Бронювання столу;

- Оформити та підтвердити замовлення;
- Зберігання історії замовлення клієнту;
- Редагування замовлення клієнта;
- Створення облікового запису клієнта.

Створений вебсайт, який буде виконувати ці функції, забезпечить для клієнта комфортне користування вебсайтом.

1.3 Постановка задачі на розробку програмного забезпечення вебсайту для ресторану «International»

Провівши аналіз предметної області та методів створення програмного забезпечення вебсайту для ресторану «International», робимо висновок, що існує необхідність створення відповідного вебсайту. Це дозволить автоматизувати процес замовлення, поліпшить огляд інформації ресторану, надасть можливість бронювання місця на певний час й дату, за допомогою комп'ютерних технологій.

Постановка задачі: розробити програмне забезпечення вебсайту для ресторану «International».

Вимоги до функціональних характеристик:

- Перехід на певні пункти сайту, через посилання у меню панелі навігації;
- При переході на сторінку меню, треба запропонувати клієнту кожну одиницю зі страв. Інформація про страву матиме зміст: назви страви, зображення, ціну, вагу, короткий опис;
- Перегляд слайдів у галереї, де клієнт може продивитися слайди приміщення ресторану;
- В описі ресторану розробити посилання на соціальні мережі;

- Додавання страви з меню у замовлення;
- Оформлення та підтвердження замовлення;
- Дати можливість забронювати місце на певний час й дату;
- Додати можливість створення облікового запису для клієнта;
- Створення та зберігання даних клієнта;
- Створення та зберігання інформації замовлення клієнта;
- Створення та зберігання елементів у меню;
- Редагування профілю клієнта;
- Редагування елементу у меню;
- Редагування замовлення клієнта.

Вимоги до організації вхідних та вихідних даних.

Вхідними даними для створення елементу у меню є зображення, та текст. Введення текстової інформації буде здійснюватися клавіатурою, а зображення можна завантажувати у форматі jpg, png у базу даних.

Вхідними даними для створення профілю буде текст, та електронна пошта. Введення тексту та електронної пошти буде здійснюватися клавіатурою.

Вхідними даними для замовлення клієнта буде додання даних з бази даних про елемент з меню у замовлення до клієнта за його ідентифікаційним номером.

Вхідними даними для бронювання столу буде панель з датою, часом, та номером столу.

2 ПРОЄКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ ДЛЯ РЕСТОРАНУ «INTERNATIONAL»

2.1 Ескізний проєкт програмного забезпечення вебсайту для ресторану «International»

2.1.1 Побудова моделі варіантів використання вебсайту «International»

Діаграма використання — це найбільш загальне уявлення функціонального призначення системи [4]. На діаграмі використання застосовуються два типу основних сутностей: варіанти використання і дійові особи, між якими встановлюються такі основні типи відносин:

- асоціація між дійовою особою і варіантом використання;
- узагальнення між варіантами використання;
- залежності (різних типів) між варіантами використання.

Головними акторами у діаграмі варіантів використання є «Клієнт» та «Менеджер», через те, що вони є у пріоритеті системи.

Короткий опис акторів представлено у таблиці 2.1.

Таблиця 2.1 – Виявлення акторів

Актори	Короткий опис
Клієнт	Має можливість оглянути сайт, оглянути меню, оформити та відредагувати замовлення, бронювання столу на певний час і дату.
Менеджер	Має можливість оглянути сайт, оглянути список усіх замовлень, змінювати меню, забронювати стіл для клієнту на певний час і дату.

Виявлення варіантів використання представлені у таблиці 2.2.

Таблиця 2.2 – Виявлення варіантів використання

Код	Основний актор	Назва	Формулювання
A1	Менеджер	Внесення змін до меню	Цей варіант використання дозволяє Менеджеру змінювати меню
A2	Менеджер	Бронювання столу для Клієнта	Цей варіант використання дозволяє Менеджеру бронювати стіл для Клієнта на певний час
A3	Менеджер	Зміна замовлення	Цей варіант використання дозволяє Менеджеру змінити замовлення Клієнта у разі відмови
K1	Клієнт	Формування замовлення	Цей варіант використання дозволяє Клієнту обрати страву із меню у замовлення
K2	Клієнт	Бронювання столу на певний час і дату	Цей варіант використання дозволяє Клієнту забронювати стіл на певний час і дату разом із замовленням
K3	Клієнт	Зміна замовлення	Цей варіант використання дозволяє Клієнту змінити своє замовлення (додати або видалити елемент із замовлення)
K4	Клієнт	Створення облікового запису	Цей варіант використання дозволяє Клієнту створити свій обліковий запис
K5	Клієнт	Огляд Меню	Цей варіант використання дозволяє Клієнту оглянути меню

Діаграму варіантів використання вебсайту для ресторану «International» наведено на рисунку 2.1.

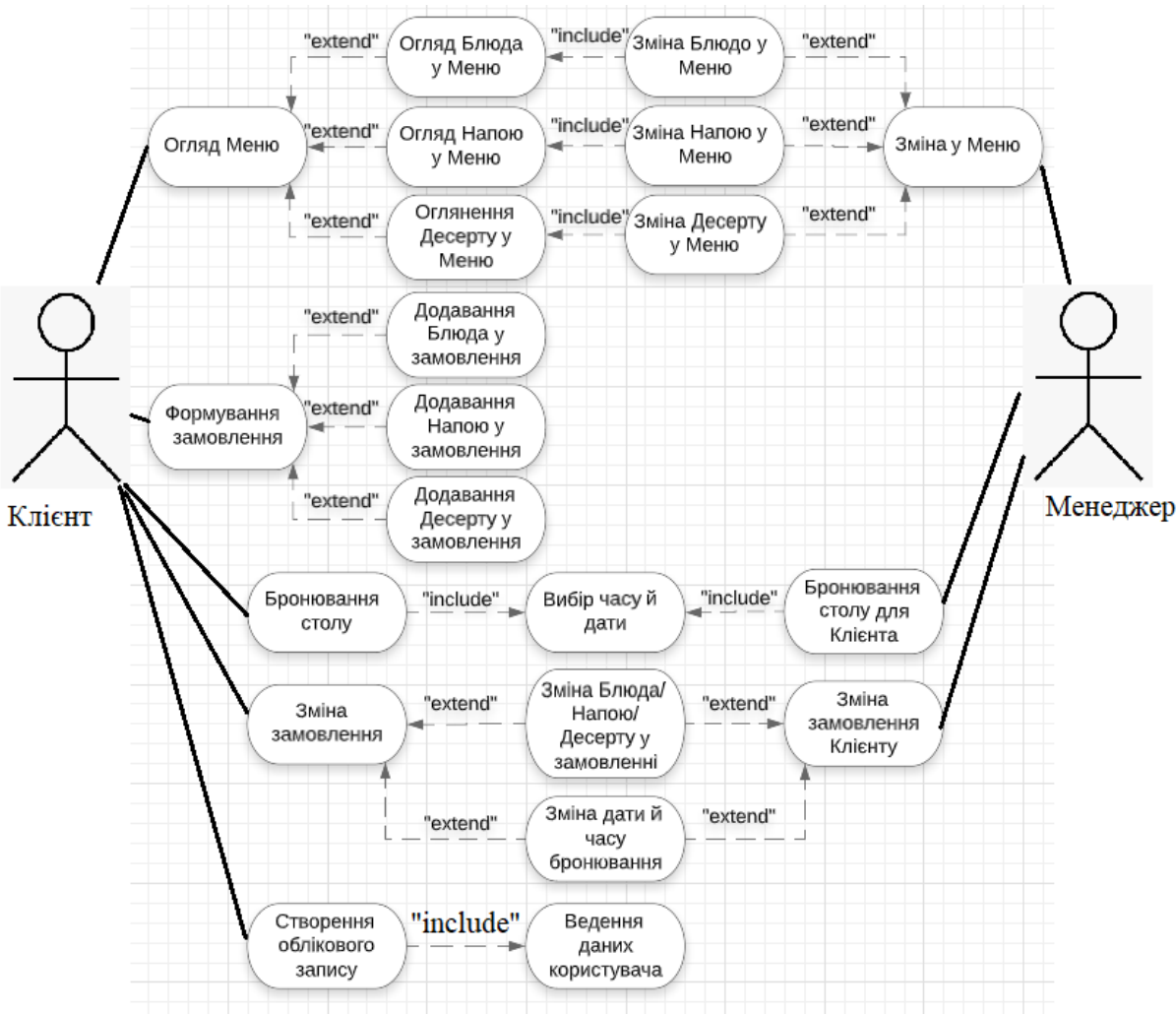


Рисунок 2.1 – Діаграма варіантів використання вебсайту «International»

2.1.2 Конкретизація варіантів використання вебсайту «International»

Внесення змін до меню від Менеджеру представлено у таблиці 2.3.

Таблиця 2.3 – Внесення змін до Меню

A1	Менеджер	Зміна у Меню	Цей варіант використання дозволяє Менеджеру змінювати меню
----	----------	--------------	--

Основна діюча особа: Менеджер.

Короткий опис:

Даний варіант використання дозволяє Менеджеру вносити зміни у меню, а саме кожного елемента: назву, ціну, вагу, короткий опис, зображення.

Бронювання столу для Клієнта наведено у таблиці 2.4.

Таблиця 2.4 – Бронювання столу для Клієнта

A2	Менеджер	Бронювання столу для Клієнта	Цей варіант використання дозволяє Менеджеру бронювати стіл для Клієнта на певний час і дату
----	----------	------------------------------	---

Основна діюча особа: Менеджер.

Короткий опис:

Даний варіант використання дозволяє Менеджеру Клієнту допомогти, або замість нього забронювати стіл на дату на певний час.

Зміна замовлення від Менеджеру представлена у таблиці 2.5.

Таблиця 2.5 – Зміна замовлення

A3	Менеджер	Зміна замовлення	Цей варіант використання дозволяє Менеджеру змінити замовлення Клієнта у разі відмови
----	----------	------------------	---

Основна діюча особа: Менеджер.

Короткий опис:

Даний варіант використання дозволяє Менеджеру вносити зміни у замовлені клієнта: видалення, додати елемент, видалити елемент.

Оформлення замовлення Клієнтом представлена у таблиці 2.6

Таблиця 2.6 – Оформлення замовлення.

K1	Клієнт	Формування замовлення	Цей варіант використання дозволяє Клієнту обрати елемент з меню у замовлення
----	--------	-----------------------	--

Діюча особа Клієнт.

Короткий опис:

Даний варіант використання дозволяє Клієнту обрати елемент з меню у замовлення самостійно, обрати страву з меню.

Бронювання столу на певний час і дату Клієнтом представлена у таблиці 2.7

Таблиця 2.7 – Бронювання столу на певний час та дату

K2	Клієнт	Бронювання столу на певний час та дату	Цей варіант використання дозволяє Клієнту забронювати стіл на певний час і дату разом із замовленням
----	--------	---	---

Діюча особа Клієнт.

Короткий опис:

Даний варіант використання дозволяє Клієнту забронювати стіл на певний час і дату.

Зміна замовлення Клієнтом наведено у таблиці 2.8

Таблиця 2.8 – Зміна замовлення

K3	Клієнт	Зміна замовлення	Цей варіант використання дозволяє Клієнту змінити своє замовлення (додати або видалити елемент із замовлення)
----	--------	---------------------	--

Діюча особа Клієнт.

Короткий опис:

Даний варіант використання дозволяє Клієнту змінити своє замовлення (додати або видалити елемент із замовлення).

Створення облікового запису Клієнтом наведено у таблиці 2.9

Таблиця 2.9 – Створення облікового запису

K4	Клієнт	Створення облікового запису	Цей варіант використання дозволяє Клієнту створити свій обліковий запис
----	--------	-----------------------------------	---

Діюча особа Клієнт.

Короткий опис:

Даний варіант використання дозволяє Клієнту створити свій обліковий запис.

Огляд меню Клієнтом наведено у таблиці 2.10

Таблиця 2.10 – Огляд меню

K5	Клієнт	Огляд Меню	Цей варіант використання дозволяє Клієнту оглянути Меню
----	--------	---------------	---

Діюча особа Клієнт.

Короткий опис:

Даний варіант використання дозволяє Клієнту забронювати стіл на певний час та дату.

Головними користувачами програмного забезпечення є Клієнт та Менеджер, тому що вони є у пріоритеті у системі.

1) Клієнт може зареєструватися у вебсайті, для цього йому треба вказати своє ім'я та пошту. Завдяки цьому клієнт зможе:

- оглянути меню;
- сформувати замовлення;
- бронювати стіл на певний час і дату;
- змінити замовлення;
- створити обліковий запис.

2) Менеджер має особливий статус, завдяки цьому Менеджер зможе:

- змінити меню;
- бронювання столу для Клієнта;
- змінити замовлення Клієнта.

2.1.3 Моделювання сценаріїв основних варіантів використання вебсайту «International» за допомогою діаграми діяльності

Призначення діаграми діяльності є моделювання процесу виконання операцій. Тому діаграма діяльності у деяких випадках схожа на діаграму станів, але різниця у тому, що на переходах між станами не вказується сигнатура подій, а також відрізняється семантика станів. За допомогою діаграми діяльності можна змоделювати необхідну для зображення алгоритму виконання операції. Сама діаграма діяльності зображує процес виконання операції із результатами діяльності на кожному з етапів [5].

Виконаємо моделювання сценаріїв роботи варіантів використання «Реєстрація, авторизації, та створення замовлення Клієнтом» за допомогою діаграми діяльності.

Діаграму діяльності варіанта використання «Реєстрація, авторизації, та створення замовлення Клієнтом» наведено на рисунку 2.2:

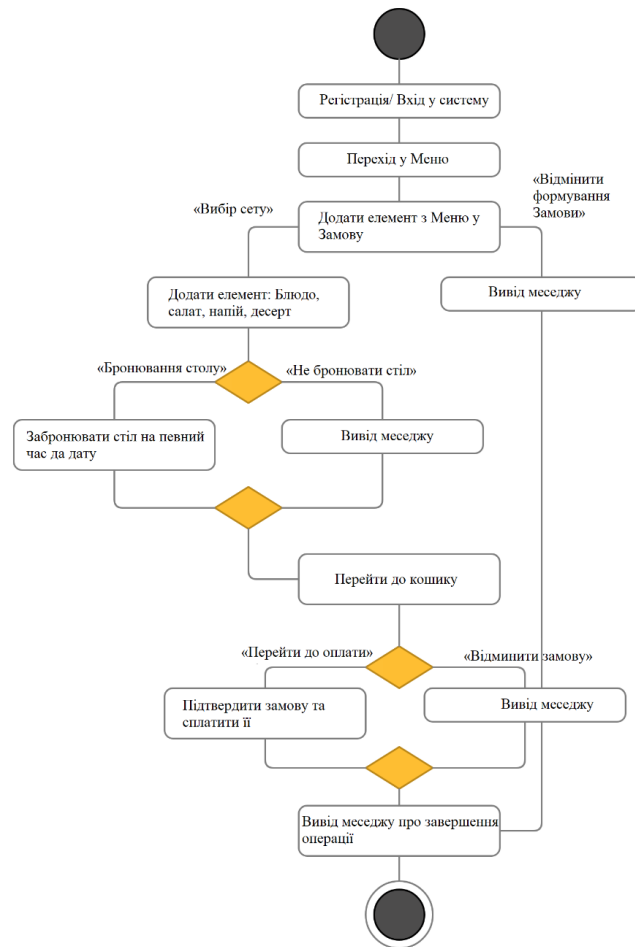


Рисунок 2.2 – Діаграма діяльності варіантів використання «Реєстрація, авторизації, та створення замовлення Клієнтом» для вебсайту «International»

2.1.4 Інформаційна модель даних вебсайту

Модель «сутність-зв'язок» — це модель даних, що дозволяє описувати концептуальні схеми у виді графічної діаграми, яка містить невелику кількість різномірних компонентів. Основними поняттями моделі «сутність-зв'язок» є сутність, зв'язок і атрибут [6].

Сутність – об'єкти, які мають однакові параметри.

Атрибут сутності – це атрибут, значення якого не може повторюватися.

Зв'язок – зв'язок та взаємодія між сутностями.

Ключ сутності – значення атрибута у сукупності для кожного екземпляра сутності, який є не однаковим.

Діаграма «Сутність-Зв'язок» вебсайту для ресторану «International» наведено на рисунку 2.3 нижче:

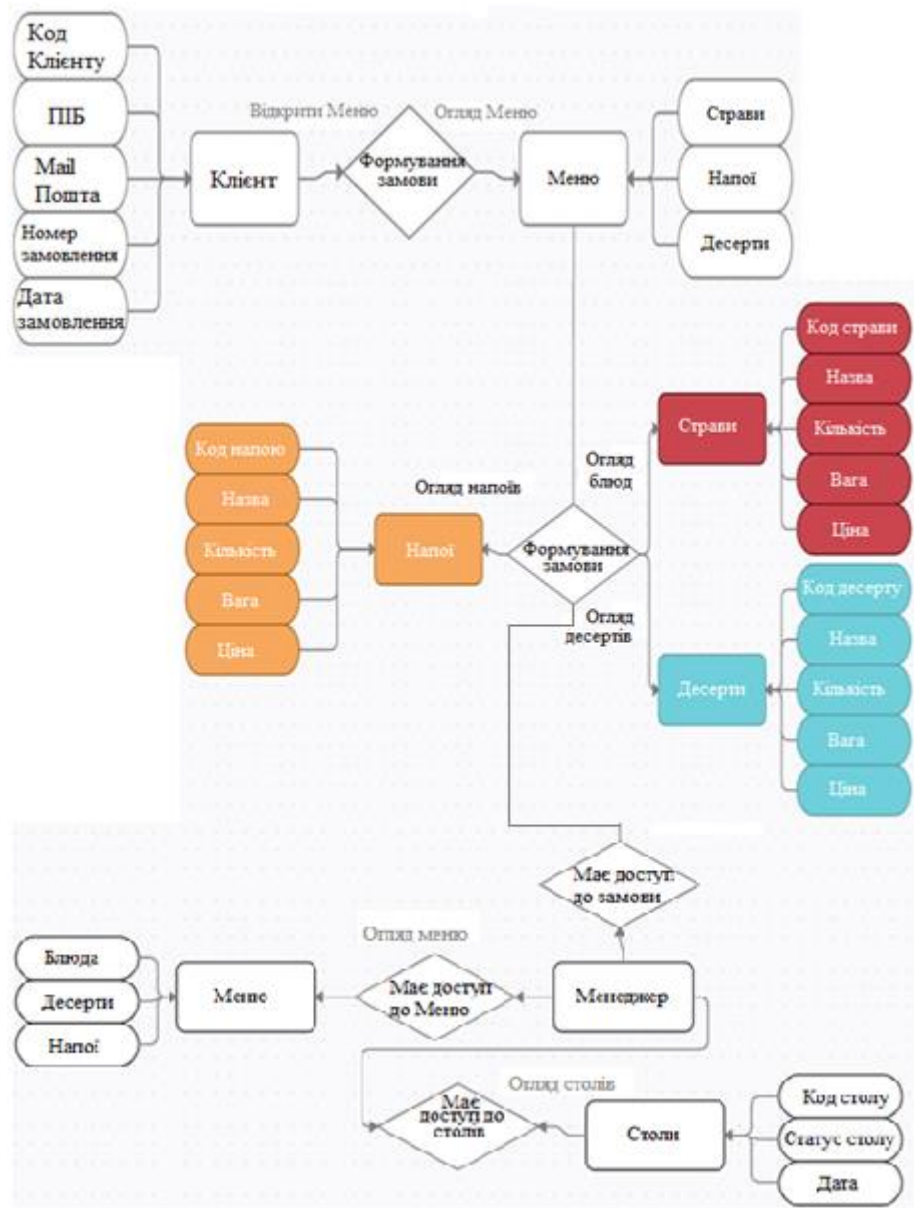


Рисунок 2.3 – Діаграма «Сутність-Зв'язок» вебсайту для ресторану «International»

2.1.5 Проєкт користувацького інтерфейсу вебсайту для ресторану «International»

Користувальницький інтерфейс – це множина програмних засобів, які забезпечують взаємодію між користувачем та комп'ютером. Головною задачею цього інтерфейсу є взаємодія діалогів. Під діалогом розуміють

взаємний обмін інформації між користувачем і програмним забезпеченням, який здійснюється у реальному часу і спрямований на вирішення завдань [7].

Було спроектовано декілька ескізів прототипів користувацького інтерфейсу програмного забезпечення вебсайту для ресторану «International» у вигляді паперових рисунків.

Паперовий варіант інтерфейс користувача головного вебсайту «International» наведено на рисунку 2.4 нижче:



Рисунок 2.4 – Паперовий варіант інтерфейс користувача головного вебсайту «International»

Паперовий варіант інтерфейс користувача опис вебсайту «International» наведено на рисунку 2.5:

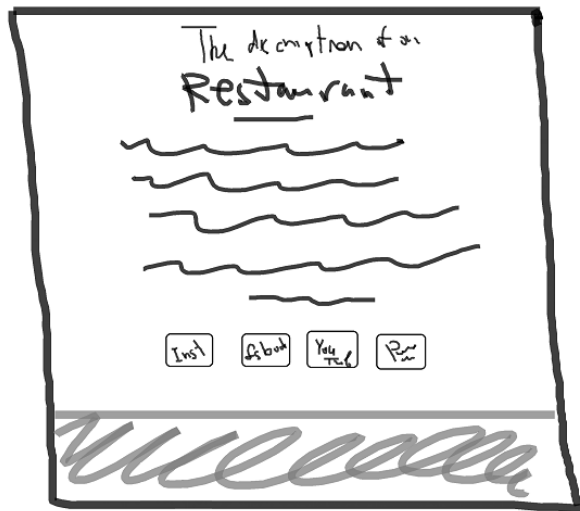


Рисунок 2.5 – Паперовий варіант інтерфейс користувача опис вебсайту
«International»

Паперовий варіант інтерфейс користувача меню вебсайту «International» наведено на рисунку 2.6:

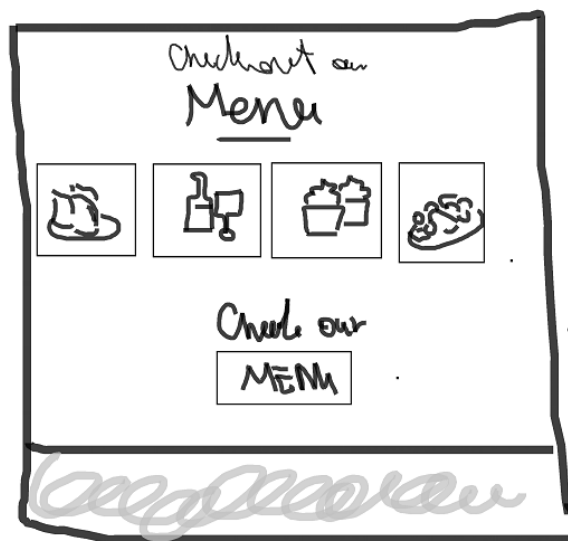


Рисунок 2.6 – Паперовий варіант інтерфейс користувача меню вебсайту
«International»

Паперовий варіант вибір стилів для вебсайту Стиль №1 «International» наведено на рисунку 2.7:

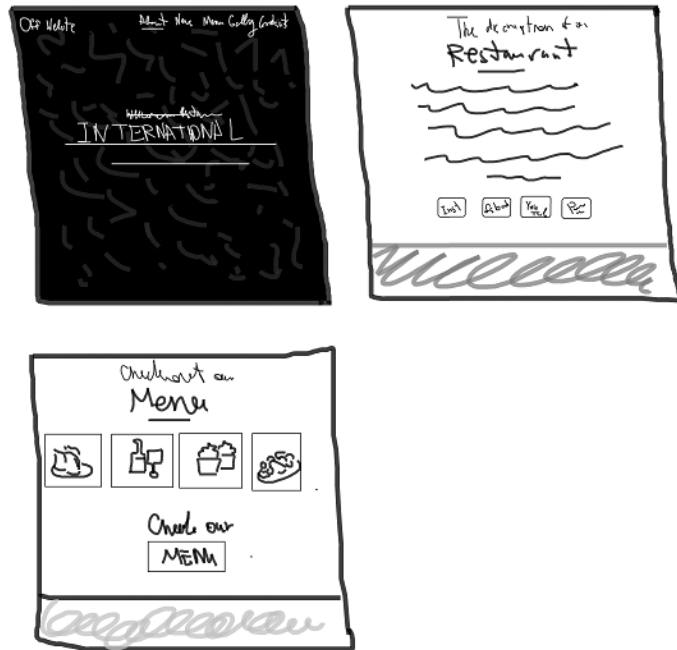


Рисунок 2.7 – Паперовий варіант вибір стилів вебсайту Стилль №1
«International»

Паперовий варіант вибір стилів для вебсайту Стилль №2 «International»
наведено на рисунку 2.8:

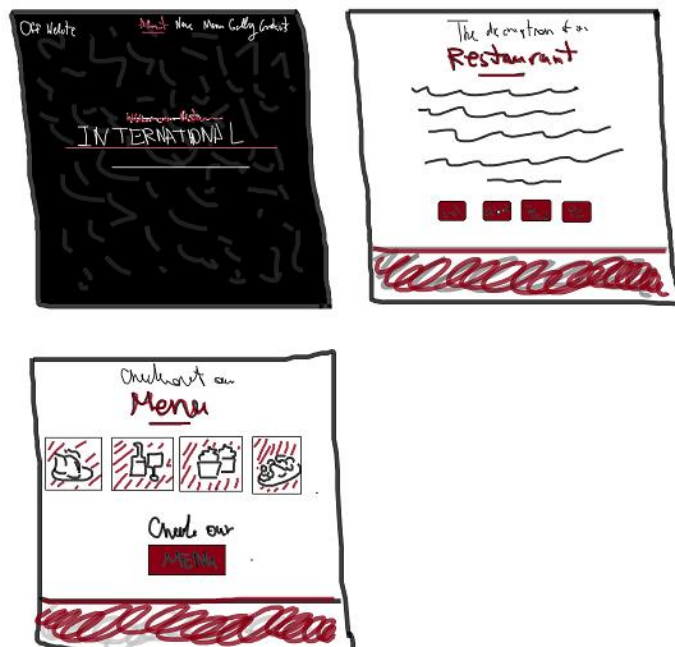


Рисунок 2.8 – Паперовий варіант вибір стилів вебсайту Стилль №2
«International»

Обґрунтуємо вибір стилів інтерфейсу користувача:

Обґрунтування кольорів:

Стиль № 1 – чорно-білий стиль.

Класичний стиль, він завжди у моді, та має свій стильовий шарм. Стиль газет, банкетного одягу.

Ці два кольори доповнюють один-одного, та не викликають у більшість людей ніякого гніту, агресії, а навпаки: спокою та гармонії, а сірий колір буде мостом між цими кольорами як логічний перехід.

Позитивне сприйняття:

Сірий колір є балансом між білим і чорним кольором. Сірий універсальний, однозначний, побутовий і консервативний, може використовуватися практично у будь-яких сферах [8].

Білий колір – це суцільний колір, асоціація чистоти та початку. Білий колір демонструє остаточне рішення [9].

Чорний колір – у практичному використанні чорний колір врівноважує білий колір. Чорний колір є класичним для костюмів, інтер'єру, дизайну автомобілів тощо [10].

Негативне сприйняття:

Сірий асоціюється з хворобами, меланхолією, депресією, поганою погодою, непотрібністю та втомою.

Занадто багато білого кольору викликає втому очей, особливо у роботі за програмою із білою темою.

Чорний колір викликає почуття важкості, страху, турботності, та асоціюється з трауром.

Стиль № 2 – червоний, білий, чорний

Стиль має благородний контраст, асоціація з античністю, романтизмом, ренесансом. Це улюблений контраст художніх майстрів. Червоний та білий колір гарно балансують та викликають пристрасть та горде почуття, а чорний колір є бар'єром, який підкреслює деталі.

Позитивне сприйняття:

Червоний колір часто використовують у дизайні одягу, автомобілів тощо [11].

Негативне сприйняття:

Червоний колір викликає роздратованість і лють. Він асоціюється з кров'ю, тобто із життям та смертю [12].

Усі шрифти бралися із сайту: «Google font» для безкоштовного використання [13].

Перелік всіх обраних шрифтів:

Приклад шрифту «Imbue» наведено на рисунку 2.9:

A black rectangular box containing the text "we had left the ground." in a light gray, lowercase, serif font.

Рисунок 2.9 – Приклад шрифту «Imbue»

Приклад шрифту «Bebas Neue» наведено на рисунку 2.10:

A black rectangular box containing the text "WE HAD LEFT THE GROUND." in a light gray, uppercase, sans-serif font.

Рисунок 2.10 – Приклад шрифту «Bebas Neue»

Приклад шрифту «Cinzel» наведено на рисунку 2.11:

A black rectangular box containing the text "WE HAD LEFT THE GROUND." in a light gray, uppercase, serif font.

Рисунок 2.11 – Приклад шрифту «Cinzel»

Приклад шрифту «Fredericka the Great» наведено на рисунку 2.12:



Рисунок 2.12 – Приклад шрифту «Fredericka the Great»

Приклад шрифту «Lora» наведено на рисунку 2.13:



Рисунок 2.13 – Приклад шрифту «Lora»

Приклад шрифту «Caveat» наведено на рисунку 2.14:

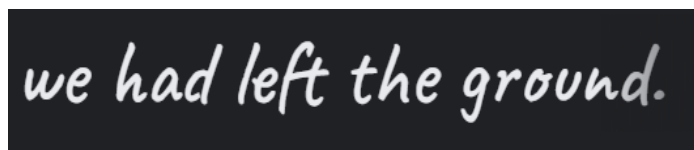


Рисунок 2.14 – Приклад шрифту «Caveat»

Обґрунтування шрифтів:

- На сайті повинен бути обов’язково головний шрифт, був обраний саме шрифт – *Imbue*. (шрифт на сайті можна буде оглянути в остаточному варіанті на рисунку 2.15).
- Вебсайт має також інформацію у текстовому форматі, текст займає 1/3 від секції на сайті, такі як: інформація, та новини. Тому для цього потрібен інший шрифт який буде допоміжний головному шрифту який ми обрали раніше. Я обрав шрифт *Lora* – він схожий на знайомий нам шрифт Times New Roman, бо такий шрифт гарно читається. (Приклад на рисунку 2.16).
- Для меню був обраний інший шрифт, а саме - *Fredericka the Great*.

(Приклад шрифт зображений на рисунку 2.17). Шрифт має класичний стиль малюнку крейдою, це додає шарму до меню.

– Шрифт для соціальних мереж краще підійде шрифт - *Caveat*. Він часто використовується на різних сайтах, бо має легкий стиль, нібито він був написаний від руки. (Приклад можна подивитися на рисунку 2.17 у надписі “Join Us”).

Остаточний варіант:



Рисунок 2.15 – Остаточний варіант головного сайту «International»

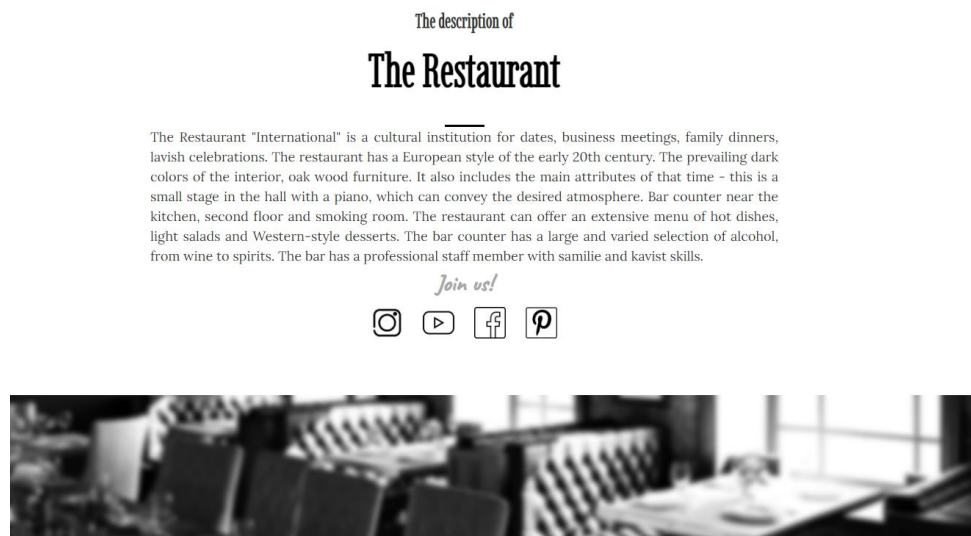


Рисунок 2.16 – Остаточний варіант опису сайту «International»

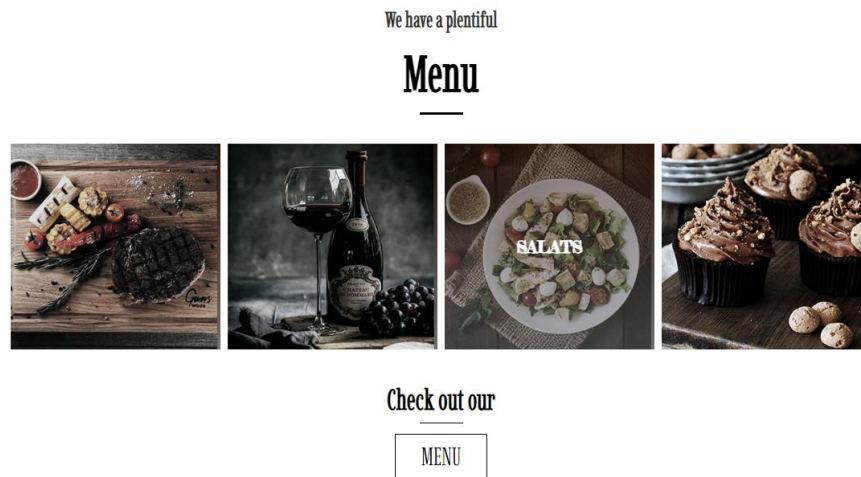


Рисунок 2.17 – Остаточний варіант Меню сайту «International».

2.2 Технічний проєкт вебсайту на базі браузера для ресторану «International»

2.2.1 Статична модель вебсайту «International»

Діаграма класів - діаграми, на якій представлена сукупність декларативних або статичних моделей класу з атрибутами [14]. Вона призначена для представлення статичної структури, яка містить: інтерфейси, пакети та зв'язки.

Клас (class) - абстрактний опис конструкції, яка використовується для функціонування груп чи частини.

Атрибут – абстрактні елементи у класі, працюють як об'єкти.

Асоціація (association) – зв'язок між двома чи більше класами.

Спадкування – спеціальні механізми, завдяки яким передаються наслідуючи характеристики чи функції від батька до сина.

Агрегація (aggregation) - асоціація, яка призначена для уявлення зв'язку між агрегатом та складовою частиною.

Композиція – зв’язок агрегації, при яких складові частини мають такий самий час життя, які потім знищуються після завершення часу життя.

За допомогою діаграми класів змоделюємо статичне представлення предметної галузі (рисунк 2.18).

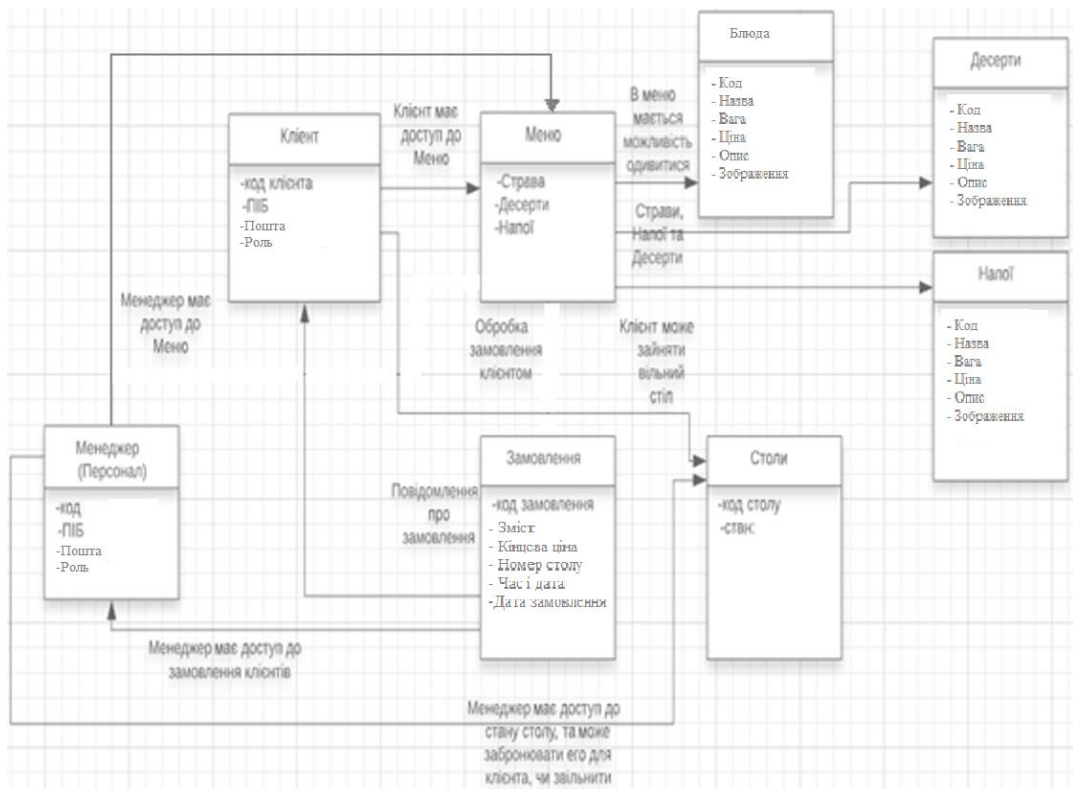


Рисунок 2.18 – діаграма класів вебсайту «International»

2.2.2 Специфікація класів з діаграми класів вебсайту «International»

- «Менеджер» - клас, що описує головного користувача, адміністратора як «Менеджер»;
- «Клієнт» - клас, що описує користувача як «Клієнт»;
- «Меню» - клас, що описує об’єкт як «Меню»;
- «Блюда» - клас, що описує об’єкт як «Блюда»;
- «Напої» - клас, що описує об’єкт як «Напої»;
- «Десерти» - клас, що описує об’єкт як «Десерти»;
- «Замовлення» - клас, що описує об’єкт як «Замовлення»;
- «Столи» - клас, що описує об’єкт як «Столи».

2.2.3 Динамічна модель програмного забезпечення вебсайту «International»

Діаграма послідовності – діаграма, на якій демонструються послідовність зв'язків між об'єктами та користувачем [15].

Лінія життя об'єкта – представляється на діаграмі послідовності у виді вертикальної лінії, яка показує існування об'єкта протягом певного проміжку часу.

Фокус управління – особливий символ на діаграмі послідовності, де об'єкт у активному стані, виконує дію й також вказується його проміжок часу.

Спроекуємо діаграму послідовності для варіантів використання «Створення замовлення Клієнтом» та «Огляд та зміна Меню Менеджером» вебсайту «International» (рисунок 2.19).

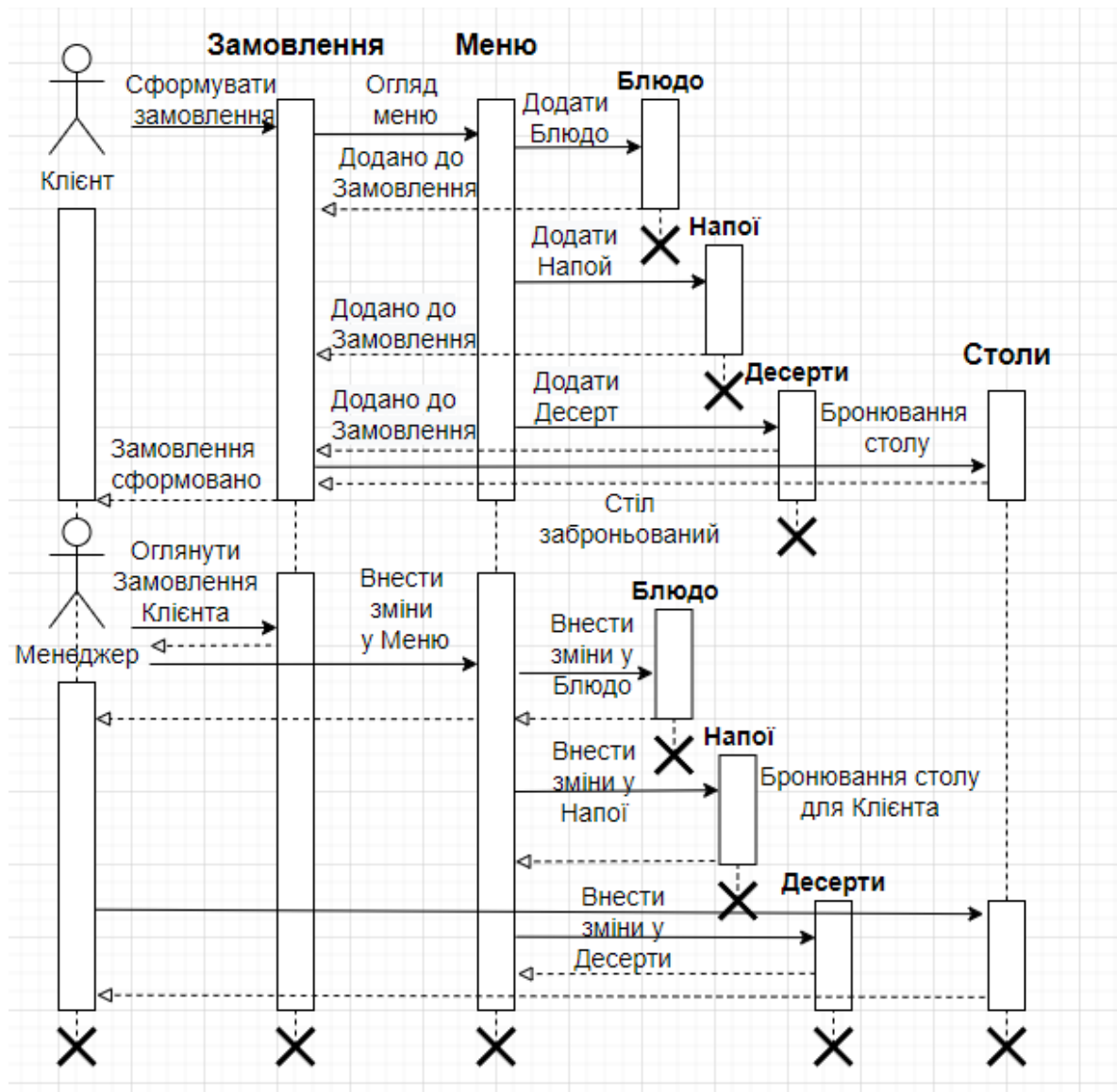


Рисунок 2.19 – Діаграма послідовності для варіантів використання «Створення замовлення Клієнтом» та «Огляд та зміна Меню Менеджером»

2.2.4 Логічна модель даних вебсайту

Логічна модель показує зв'язки між об'єктами даних. До логічного проєктування відноситься опис характеристик тих об'єктів, відомості про яких накопичуватимуться і використовуватися в програмному середовищі [16].

Для реалізації бази даних для проєкту ресторан «International» необхідно створити ряд таблиць:

- «Клієнт»;

- «Меню»;
- «Замовлення»;
- «Менеджер»;
- «Блюда»;
- «Десерти»;
- «Напої»;
- «Столи».

Частина цих таблиць є основними, тому, що вони відтворюють процеси, які відбуваються. У цих таблицях створені поля, кожне з яких зустрічається в інших, відповідних їм таблицях. Ці таблиці містять інформацію для формування звітів. Такими є:

- «Фінансовий звіт»;
- «Звіт про стан столу».

Деякі таблиці являються допоміжними або довідниками.

- «Клієнт»;
- «Менеджер»;
- «Меню»;
- «Замовлення»;
- «Столи».

Логічна модель даних вебсайту «International» представлена на рисунку 2.20.

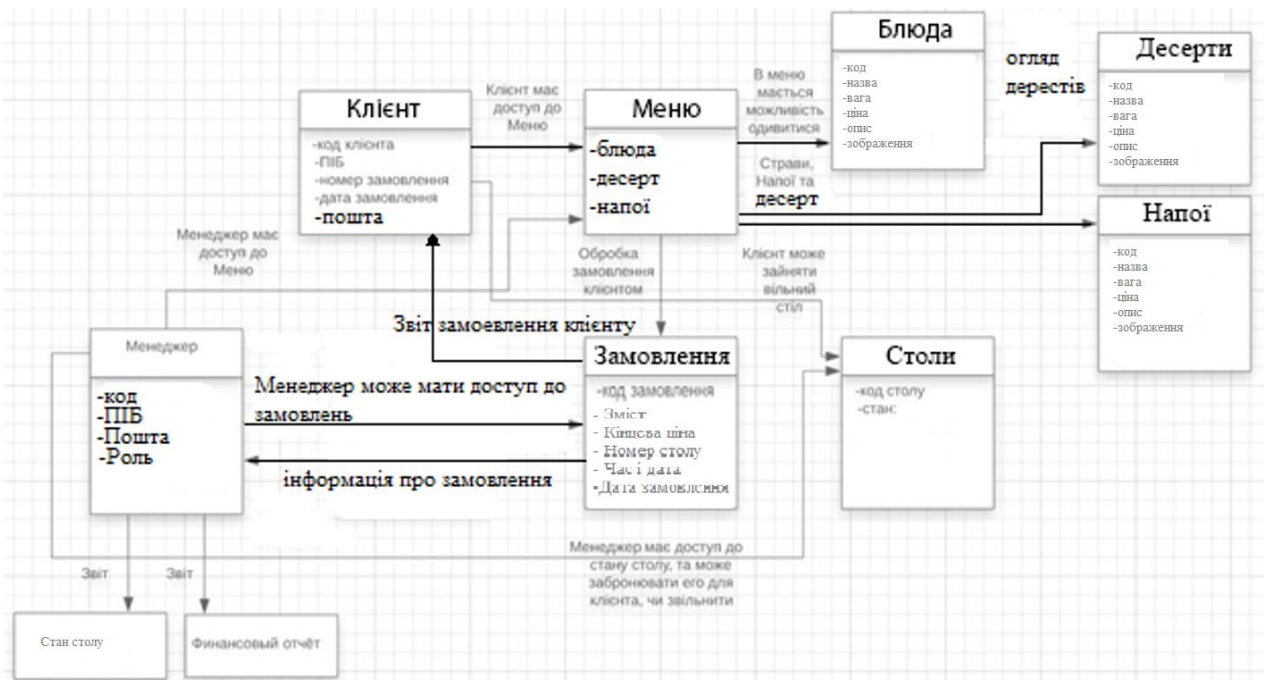


Рисунок 2.20 – Логічна модель даних до вебсайту «International»

2.3 Робочий проєкт вебсайту «International»

2.3.1 Обґрунтування вибору засобів програмування

Для реалізації проєкту було обрано ряд технологій. Для frontend є велика кількість фреймворків, мій вибір зупинився на VueJS [17]. Цей фреймворк має величезну популярність і потрібність на ринку. Фреймворк VueJS прийшов на заміну таким старим фреймворкам як Angular і J-Query та іншим.

Оскільки фреймворк VueJS має синтаксис JavaScript [18], тоді я використовуватиму саме цю основну мову програмування для створення вебсайту. Для створення шаблону сайту використовуватиму HTML [19], а для стилістики CSS [20]. Представлені технології буде цілком достатньо для створення базового вебсайту, а для реалізації складних функцій буде залучений фреймворк.

Також вебсайту буде необхідна база даних (backend), з якою буде взаємодіяти вебсайт для зберігання даних. Оптимальним варіантом буде СКБД Firebase – вона проста у використанні [21]. Firebase – хмарна СКБД класу

NoSQL, що дозволяє розробникам вебсайтів зберігати та синхронізувати дані між декількома клієнтами, вона доступна, легко встановлюється до фреймворку, у рамках безкоштовного доступу може реалізувати: авторизацію, зберігання даних та інше.

За підсумком я використовуватиму:

Frontend:

- Vue JS 2.8.3
- JavaScript ES6
- HTML 5
- CSS 3

Backend:

- FireBase 9.6.6.
- Vue JS 2.8.3

2.3.2 Фізична модель даних вебсайту

Фізична модель даних — модель, що описує збереження даних у базі даних. Фізична модель даних залежить від СКБД, та вона базується на структурі таблиці в базі даних [22].

Для реалізації бази даних для проєкту ресторан «International» необхідно створити ряд таблиць [23]:

- «Клієнт»;
- «Меню»;
- «Замовлення»;
- «Менеджер»;
- «Блюда»;
- «Десерт»;
- «Напої»;
- «Столи».

Фізична модель даних зображена у виді таблиці 2.11.

Таблиця 2.11 – Таблиця фізичної моделі бази даних

Назва таблиці	Логічне ім'я атрибута	Фізичне ім'я атрибута	Тип даних
Клієнт	код	id	int
	ім'я	name	varchar
	пошта	email	varchar
	роль	role	varchar
Меню	код	id	int
Блюда	код	id	int
	назва	name	varchar
	вага	weight	int
	ціна	price	int
	опис	description	varchar
	зображення	image	varchar
Напій	код	id	int
	назва	name	varchar
	вага	weight	int
	ціна	price	int
	опис	description	varchar
	зображення	image	varchar

Продовження таблиці 2.11

Назва таблиці	Логічне ім'я атрибута	Фізичне ім'я атрибута	Тип даних
Десерт	код	id	int
	назва	name	varchar
	вага	weight	int
	ціна	price	int
	опис	description	varchar
	зображення	image	varchar
Замовлення	код	id	int
	дата замовлення	date	Date (yyyy:mm:dd)
	зміст замовлення	content	varchar
	номер столу	number	int
	дата бронювання	orderData	Date (yyyy:mm:dd)
	час бронювання	orderTime	Date (HH:mm)
	кінцева ціна	totalPrice	int
Менеджер	код	id	int
	ім'я	name	varchar
	пошта	email	varchar
	роль	role	varchar
Столи	код	id	int
	статус	status	varchar

2.3.3 Розробка діаграми розгортання вебсайту

Діаграма розгортання — діаграми у моделі UML, на якій зображаються обчислювальні вузли під час роботи програми компонентів, та об'єкти, які задаються у вузлах. Ці компоненти відповідають уявленню робочих екземплярів. Компоненти, у яких відсутнє уявлення робочих екземплярів при роботі програми, то такі компоненти на діаграмах не зображаються, але їх можна зобразити на діаграмах компонентів. Діаграма розгортання показує робочі екземпляри компонентів, а діаграма компонента зображає зв'язки між типами компонентів [24].

Діаграму розгортання для вебсайту представлено на рисунку 2.21.

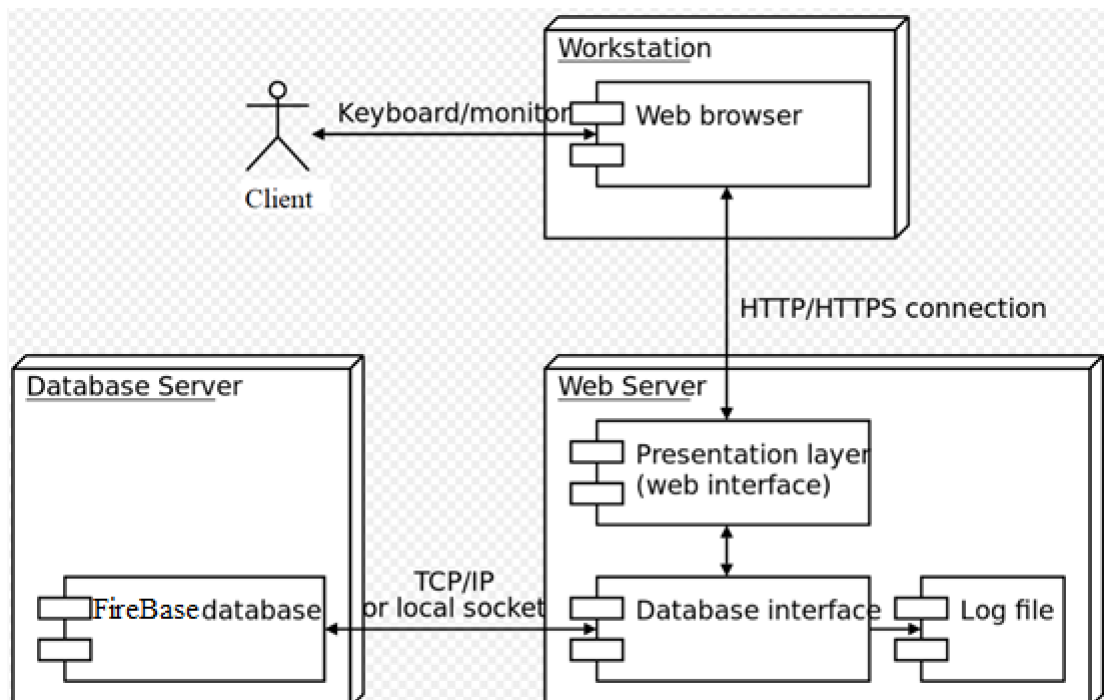


Рисунок 2.21 – Діаграма розгортання для вебсайту

2.3.4 Реалізація основних класів вебсайту

Спочатку створюється HTML файл, саме цей файл зображає загальну картину нашого проекту, а наступні файли (компоненти), будуть переводитися з мови JavaScript у мову HTML, щоб зобразити результат роботи на екран. Наступним кроком буде виклик у HTML файлі головного файлу – App.js, через

який ми будемо звертатися до інших файлів (компонентів). Щоб спростити звертання до інших файлів (компонентів), буде створено маршрутизацію до потрібних компонентів, завдяки іншого файлу – `route.js`, який і буде викликати наш головний файл `App.js`. Файл `route.js` буде перемикає компоненти програми, звертаючись до потрібного посилання (наприклад: звертання до сторінки «/home» звернеться до компонента `Home.js`, та й відкриє на вебсайті головну сторінку, а звертання до сторінки «/profile» відкриє нам сторінку профілю і так далі.), а головний файл буде відкривати той компонент, до якого ми хочемо звернутися, водночас файл HTML буде зображати результат роботи у браузері.

Реалізація коду програмного забезпечення вебсайту «International» представлено у додатку Г.

2.3.5 Тестування програмного забезпечення вебсайту «International»

Тестування програмного забезпечення – це спосіб перевірки програмного забезпечення з підготовленими результатами, для виявлення програмних помилок. Тестування полягає в опрацюванні алгоритму роботи програмного забезпечення та його послідовність. Тести підбирають так, щоб вони виконували тестування із різними типами можливих ситуацій [25].

Кінцевою метою тестування є отримання сертифіката на розроблений програмний продукт, та виявлення програмних помилок у програмному забезпеченні.

Методи тестування й верифікації залежать від методів проєктування та стадій розробки програмного забезпечення, з яких починається перевірка правильності функціонування результатів проєктування.

Статичні методи використовуються під час проведення тестування та аналізу специфікацій компонентів, а динамічні методи використовуються у процесі виконання програми [25].

Верифікація – перевірка відповідності реалізації системи специфікаціям результатів проєктування й опису компоненти,

Валідація – перевірка програмного забезпечення за ознаками вимог та задач замовника. Валідація показує, що програмне забезпечення є коректною реалізацією поставлених задач й проводиться після завершення етапу розробки програмного забезпечення [25].

Тестування.

Тестування показує реальні результати програмного продукту. Всі тестування будуть записані у таблицю тестів.

Будуть проведені наступні види тестування:

- Реєстрація;
- Авторизація;
- Додати елемент у меню;
- Видалити елемент з меню;
- Додати елемент з меню у замовленні;
- Видалити елемент із замовлення;
- Підтвердити замовлення;
- Переглянути історію замовлення.

Результат виконання тестування реєстрації, та авторизації користувача наведено у таблиці 2.12.

Таблиця 2.12 – Тест Реєстрації, та Авторизації.

№ тесту	Дія тесту	Вхідні дані	Вихідні дані	Результат тестування
1	Реєстрація	Електрона пошта; Пароль; Ім'я.	Створення нового користувача у базі даних.	Тест пройдено
	Авторизація	Електрона пошта; Пароль.	Вхід у систему.	Тест пройдено

Результат виконання тестування додавання, та видалення елементу з Меню наведено у таблиці 2.13

Таблиця 2.13 – Тест додавання, та видалення елементу з Меню.

№ тесту	Дія тесту	Вхідні дані	Вихідні дані	Результат тестування
3	Додавання елементу у Меню	Введення даних нового елементу	Додання елементу у Меню	Тест пройдено
	Видалення елементу з Меню	Запит на видалення елементу з Меню	Відсутність елементу у Меню	Тест пройдено

Результат виконання тестування додання та видалення елементу з меню у замовлення наведено у таблиці 2.14.

Таблиця 2.14 – Тест додання та видалення елементу з меню у замовлення.

№ тесту	Дія тесту	Вхідні дані	Вихідні дані	Результат тестування
4	Додати елемент з Меню у Замовлення	Перенос даних елементу з Меню у Замовлення	Додання елементу у Замовлення	Тест пройдено
	Видалення елементу із Замовлення	Запит на видалення елементу з Замовлення	Відсутність елементу у Замовленні	Тест пройдено

Результат виконання тестування підтвердження і перегляд замовлення наведено у таблиці 2.15

Таблиця 2.15 – Тест підтвердження і перегляд замовлення.

№ тесту	Дія тесту	Вхідні дані	Вихідні дані	Результат тестування
5	Підтвердження замовлення	Запит на підтвердження, та перенос даних із кошику у замовлення	Дані у базі даних замовлення	Тест пройдено
	Перегляд історії замовлення	Перехід у сторінку «Історія замовлення»	Інформація у історії замовлення	Тест пройдено

Валідація.

Було передбачено, та запропоновано наступні валідації:

- Коректне ведення електронної пошти;
- Не залишати поля пустими;
- Не перевищувати кількість символів;
- Задавати більшу кількість символів.

Приклад, роботи валідації сторінку логіну, можна побачити на рисунках 2.22, 2.23, 2.24.

Домашняя бухгалтерия

Email

Пароль

Забыли пароль?

ВОЙТИ >

Нет аккаунта? **ЗАРЕГИСТРИРУЙТЕСЬ**

Рисунок 2.22 – Пример валидации логина

Домашняя бухгалтерия

Email

Поле не должно быть пустым

Пароль

Введите пароль

Забыли пароль?

ВОЙТИ >

Нет аккаунта? **ЗАРЕГИСТРИРУЙТЕСЬ**

Рисунок 2.23 – Пример валидации логина

Домашняя бухгалтерия

Email

Пароль

Имя

☐ С правилами согласен

ЗАРЕГИСТРИРОВАТЬСЯ ➤

Уже есть аккаунт? **ВОЙТИ!**

Рисунок 2.26 – Пример валидации регистрации

Домашняя бухгалтерия

Email

Поле не должно быть пустым

Пароль

Введите пароль

Имя

Введите ваше имя

☐ С правилами согласен

ЗАРЕГИСТРИРОВАТЬСЯ ➤

Уже есть аккаунт? **ВОЙТИ!**

Рисунок 2.27 – Пример валидации регистрации

Домашняя бухгалтерия

Email

test

Введите корректный Email

Пароль

.....

Имя

Pavel

☐ С правилами согласен

ЗАРЕГИСТРИРОВАТЬСЯ ➤

Уже есть аккаунт? **ВОЙТИ!**

Рисунок 2.28 – Приклад валідації реєстрації

2.3.6 Випробування вебсайту

Випробування програмного забезпечення визначить реальні результати програмного забезпечення, та показ реалізації за технічним завданням. У випадку виявленні помилок, програмне забезпечення буде допрацьовуватися, буде проведено повторно тести.

Виконаємо випробування програми від лиця Клієнта, та Менеджера. Проведемо повний тест випробування програмного забезпечення. Результати показані у виді знімку з екрану та із описом дій.

Результат вхід на сторінку сайту показано на рисунку 2.29.



Рисунок 2.29 – Вхід до головної сторінки вебсайту «International»

Реєстрація користувача наведена на рисунку 2.30.

Рисунок 2.30 – Реєстрація користувача

Результат авторизація користувача наведена на рисунку 2.31.

Домашня бухгалтерия

Email

test@test.com

Пароль

.....|

Забыли пароль?

ВОЙТИ >

Нет аккаунта? ЗАРЕГИСТРИРУЙТЕСЬ

Рисунок 2.31 – Авторизация користувача

Результат входу у систему користувачем наведена на рисунку 2.32.



Рисунок 2.32 – Результат входу у систему

Результат входу у Меню наведена на рисунку 2.33.

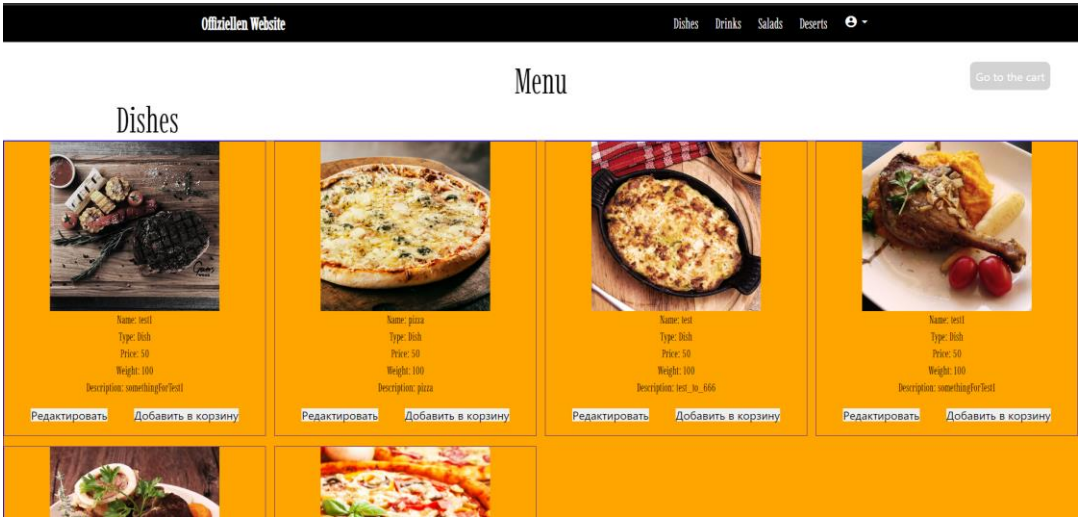


Рисунок 2.33 – Вхід до Меню

Перевірка кошику замовлення до додавання товару наведена на рисунку 2.34.

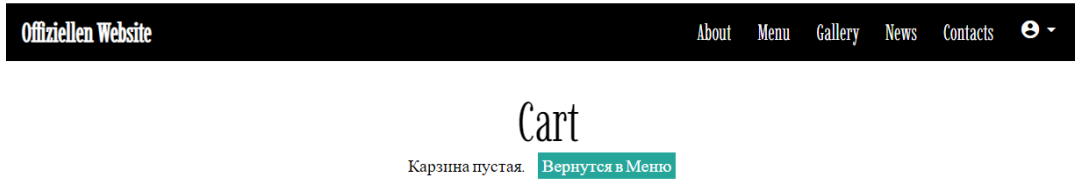


Рисунок 2.34 – Перевірка кошику замовлення

Перевірка замовлення до підтвердження наведена на рисунку 2.35.



Рисунок 2.35 – Перевірка замовлення

Перегляд кошика після додавання декількох елементів у Корзину наведено на рисунку 2.36.

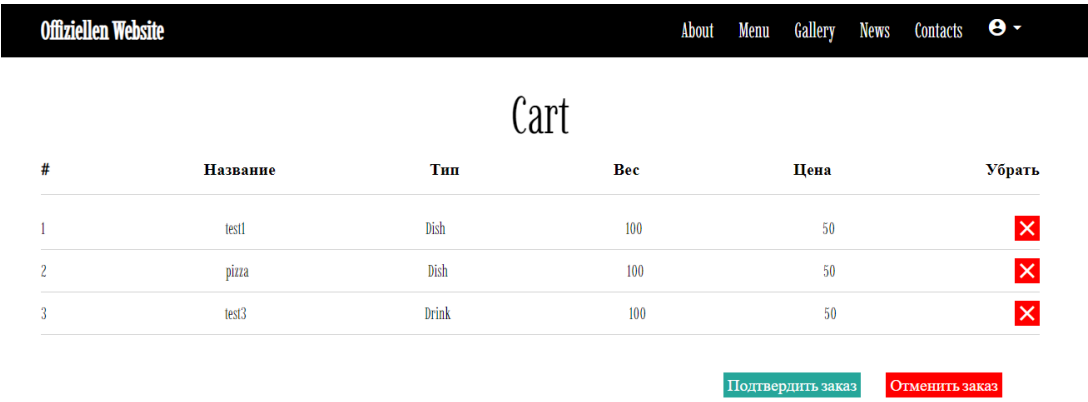


Рисунок 2.36 – Сторінка формування замовлення з елементами

Видалення елемента із замовлення, наведено на рисунку 2.37.

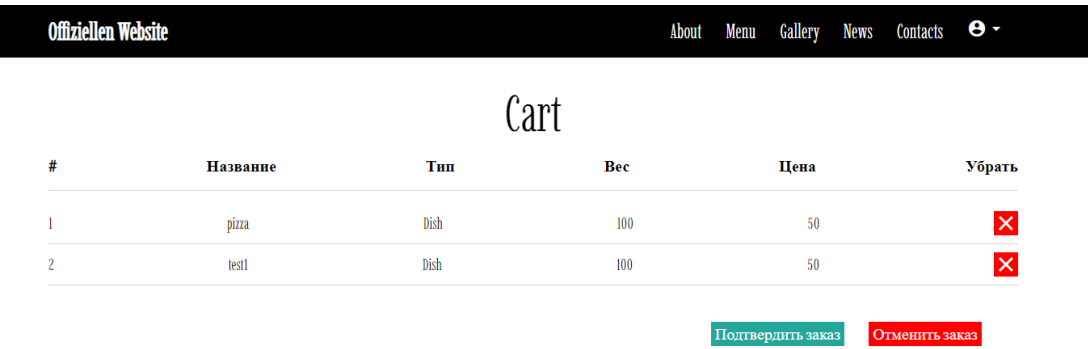


Рисунок 2.37 – Видалення елемента із замовлення

Перегляд Замовлення після підтвердження наведено на рисунку 2.38, 2.39.

Offiziellen Website

AboutMenuGalleryNewsContacts

Order

Заказ №25268

Наименование:	test1,test3
Стоимость:	50,50
Итоговая цена:	100 €
Дата:	10.06.2022

Редактировать

Заказ №30625

Наименование:	pizza,Цезарь
Стоимость:	50,50
Итоговая цена:	100 €
Дата:	10.06.2022

Редактировать

Рисунок 2.38 – Огляд списку замовлення

Редактировать

Заказ №22225

Наименование:	test1,pizza,test1,test3,Цезарь,test2
Стоимость:	50,50,50,50,50,50
Итоговая цена:	300 €
Дата:	11.06.2022

Редактировать

Заказ №98865

Наименование:	test1,pizza,test3
Стоимость:	50,50,50
Итоговая цена:	150 €
Дата:	11.06.2022

Редактировать

Рисунок 2.39 – Огляд списку Замовлення

Сторінка вебсайту Адміністратора наведено на рисунку 2.40.

Offiziellen WebsiteAdmin

Home

Список Меню

Заказы

Список Клиентов

HOME

Рисунок 2.40 – Сторінка Адміністратора

Сторінка Адміністратора додавання елементу у Меню наведено на
рисунку 2.41.

Offiziellen Website

Admin

TEST

Current User

Name: Admin

Role: admin

1

New Element Menu

Выберите категорию

Dish

Название

test1

Вес

100

Цена

50

Описание

somethingForTest1

Загрузить

Изображение

СОЗДАТЬ

Рисунок 2.41 – Додавання елементу у Меню

3 РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ ДЛЯ РЕСТОРАНУ «INTERNATIONAL»

Результатом розробки програмного забезпечення вебсайту для ресторану «International» була реалізація функціональних характеристик, таких як додавання страв з меню у замовлення, оформлення та підтвердження замовлення, бронювання місця на певний час й дату, створення облікового запису для клієнта, створення та зберігання елементів у меню, редагування профілю клієнта, редагування елемента у меню та редагування замовлення клієнта.

Процес розробки включав аналіз предметної області та аналіз існуючих вебсайтів ресторанів. Завдяки аналізу аналогів виявлено їх недоліки, переваги, та завдяки цьому виконано постановку задачі на розробку програмного забезпечення вебсайту.

В ескізному проєкті побудовано діаграму варіантів використання та діаграму діяльності, спроектовано інформаційну модель даних «Сутність-зв'язок», а також спроектовано інтерфейс користувача.

У технічному проєкті змодельовано діаграму послідовності. Також було спроектовано статичну і логічну модель.

У робочому проєкті обґрунтовано вибір мови програмування та СКБД. Спроектовано фізичну модель даних та діаграму розгортання. А також було проведено тестування програмного забезпечення. Результати тестування програмного забезпечення можна переглянути у пункті 2.3.5.

Останнім етапом було випробування програмного забезпечення за програмою та методикою випробувань. Результати випробувань було представлено у пункті 2.3.6, вони показали правильність роботи програмного забезпечення вебсайту для ресторану «International».

4 ОХОРОНА ПРАЦІ

4.1 Загальні вимоги безпеки [26]

4.1.1 На програміста можуть впливати небезпечні й шкідливі виробничі фактори:

А) фізичні:

- підвищений рівень електромагнітного випромінювання;
- підвищені рівні рентгеновського випромінювання;
- підвищені рівні ультрафіолетового випромінювання;
- підвищений рівень інфрачервоного випромінювання;
- підвищений рівень статичної електрики;
- підвищені рівні запиленості повітря робочої зони;
- знижена або підвищена вологість повітря робочої зони;
- знижена або підвищена рухливість повітря робочої зони;
- підвищений рівень шуму;
- підвищений або знижений рівень освітленості;
- нерівномірність розподілу яскравості в полі зору;
- підвищена яскравість світлового зображення;
- підвищений рівень пульсації світлового потоку;
- підвищене значення напруги в електричному ланцюзі, замикання якого може відбутися через тіло людини;

Б) хімічні:

- підвищений вміст в повітрі робочої зони двоокису вуглецю, озону, аміаку, фенолу, формальдегіду і поліхлорованому біфенілі;

В) психофізіологічні:

- напруга зору;
- напруга уваги;
- інтелектуальні навантаження;
- емоційні навантаження;
- тривалі статичні навантаження;
- монотонність праці;
- великий обсяг інформації, що обробляється в одиницю часу;
- нераціональна організація робочого місця;

Г) біологічні:

- підвищений вміст в повітрі робочої зони мікроорганізмів.

4.1.2 До робіт програмістом допускаються:

- особи не молодше 18 років, що пройшли обов'язковий при прийманню на роботу і щорічні медичні огляди на предмет придатності для роботи;
- пройшли вступний інструктаж з охорони праці;
- пройшли навчання безпечним приймання і методам праці за програмою, затвердженою керівником підприємства (роботодавцем), розробленої на основі Типової програми, і пройшли перевірку знань, в тому числі з електробезпеки;
- пройшли курс навчання на персональному комп'ютері з використанням конкретного програмного забезпечення;
- які пройшли інструктаж з охорони праці на конкретному робочому місці по даній інструкції.

4.1.3 Програміст повинен бути забезпечений ЗІЗ відповідно до міжгалузевих правил забезпечення працівників спеціальним одягом, спеціальним взуттям та іншими засобами індивідуального захисту, видані працівникам засоби індивідуального захисту повинні відповідати характеру й умовам роботи й забезпечувати безпеку праці. Не допускаються придбання і

видача працівникам засобів індивідуального захисту без сертифіката відповідності. Характеристика виданих ЗІЗ (номенклатура, термін видачі та норми відповідності) встановлюється з особистих карток працівників, зайнятих на певному робочому місці. Нормативні номенклатура і терміни видачі ЗІЗ визначаються відповідно до Типових галузевих норм безплатної видачі робітникам і службовцям спеціального одягу, спеціального взуття та інших засобів індивідуального захисту.

4.2 Вимоги безпеки перед початком роботи

4.2.1 Перед початком роботи програміст зобов'язаний:

- оглянути й давати (робити) лад робоче місце;
- відрегулювати освітленість на робочому місці, переконатися в достатності освітленості, відсутності відбивання на екрані, відсутності зустрічного світлового потоку;
- перевірити правильність підключення обладнання в електромережу;
- протерти спеціальною серветкою поверхню екрана;
- переконатися у відсутності дискет у дисководі процесора персонального комп'ютера;
- перевірити правильність установки столу, стільця, підставки для ніг, пюпітра, положення обладнання, кута нахилу екрана, положення клавіатури та, при необхідності, провести регулювання робочого столу і крісла, а також розташування елементів комп'ютера відповідно до вимог ергономіки та з метою виключення незручних поз.

4.2.2 При включенні комп'ютера дотримуватися правил електробезпеки.

4.2.3 Програмісту забороняється приступати до роботи при:

- відсутності на ВДТ гігієнічного сертифіката, що включає оцінку візуальних параметрів;
- відсутності інформації про результати атестації умов праці на даному робочому місці або при наявності інформації про невідповідність параметрів даного обладнання вимогам санітарних норм;
- відсутності захисного екранного фільтра класу "повний захист";
- виявленні несправності обладнання;
- відсутності захисного заземлення пристроїв ПЕОМ і ВДТ;
- відсутності вуглекислотного або порошкового вогнегасника та аптечки першої допомоги;
- порушення гігієнічних норм розміщення ВДТ (при однорядному розташуванні менш як 1 м від стін, при розташуванні робочих місць в колону на відстані менш як 1,5 м, при розміщенні на площі менш як 6 кв. м на одне робоче місце, при рядну розміщенні дисплеїв екранами один до одного).

4.3 Вимоги безпеки під час виконання роботи

4.3.1 Програміст під час роботи зобов'язаний:

- виконувати тільки ту роботу, яка йому була доручена і по якій він був проінструктований;
- протягом усього робочого дня тримати в порядку і чистоті робоче місце;
- тримати відкритими всі вентиляційні отвори пристроїв;
- при необхідності припинення роботи на деякий час закрити всі активні завдання;
- виконувати санітарні норми й дотримуватися режими роботи й відпочинку;
- дотримуватися правил експлуатації обчислювальної техніки відповідно до інструкцій по експлуатації;

- дотримуватися встановлених режимом робочого часу регламентовані перерви в роботі й виконувати рекомендовані вправи для очей, шиї, рук, тулуба, ніг;

- дотримуватися відстань від очей до екрана в межах 60 - 80 см.

4.3.2 Програмісту під час роботи забороняється: торкатися до задньої панелі системного блоку (процесора) при включеному живленні; перемикає роз'єми інтерфейсних кабелів периферійних пристроїв при включеному живленні; захащувати верхні панелі пристроїв паперами та сторонніми предметами; допускати захащеність робочого місця папером - в цілях недопущення накопичення органічного пилу; Проводити відключення живлення під час виконання активної задачі; Проводити часті перемикання харчування; допускати попадання вологи на поверхню системного блоку (процесора), монітора, робочу поверхню клавіатури, дисководів, принтерів та ін. пристроїв; включати сильно охолоджене (принесене з вулиці в зимовий час) обладнання; Проводити самостійно розкриття і ремонт обладнання.

4.4 Вимоги безпеки в аварійних ситуаціях

4.4.1 Програміст зобов'язаний:

- у всіх випадках виявлення обриву проводів живлення, несправності заземлення та інших пошкоджень електрообладнання, появи запаху гару негайно відключити живлення і повідомити про аварійну ситуацію керівнику і черговому електрику;

- при виявленні людини, що потрапила під напругу, негайно звільнити його від дії струму шляхом відключення електроживлення і до прибуття лікаря надати потерпілому першу медичну допомогу;

- при будь-яких випадках збою в роботі технічного обладнання або програмного забезпечення негайно викликати представника інженерно-технічної служби експлуатації обчислювальної техніки;

- в разі появи різі в очах, при різкому погіршенні видимості - неможливості сфокусувати погляд або навести його на різкість, появі болю в пальцях і кистях, посилення серцебиття негайно покинути робоче місце, повідомити про те, що сталося, керівника робіт і звернутися до лікаря;

- при загорянні обладнання відключити харчування і вжити заходів до гасіння вогнища пожежі за допомогою вуглекислотного або порошкового вогнегасника, викликати пожежну команду і повідомити про подію керівнику робіт.

4.5 Вимоги безпеки після закінчення роботи

4.5.1 Після закінчення робіт програміст зобов'язаний дотримуватися наступну послідовність виключення обчислювальної техніки:

- провести закриття всіх активних завдань;
- виконати стоянку голівки, що зчитує жорсткого диска (якщо не передбачена автоматична стоянка головки);
- переконатися, що в дисководі немає дискет;
- вимкнути живлення системного блоку (процесора);
- вимкнути живлення всіх периферійних пристроїв;
- відключити блок живлення.

ВИСНОВКИ

У кваліфікаційній роботі було вирішено практичну задачу програмної інженерії, що характеризується комплексністю та невизначеністю умов, а саме розроблено програмне забезпечення вебсайту для ресторану «International».

Досягнуто кінцевий результат роботи, а саме:

- проведено аналіз предметної області розробки програмного забезпечення вебсайту;
- виконано постановку задачі;
- розроблено ескізний проєкт програмного забезпечення вебсайту;
- розроблено технічний проєкт програмного забезпечення вебсайту;
- розроблено робочий проєкт програмного забезпечення вебсайту;
- виконано кодування та тестування програмного забезпечення вебсайту;
- розглянуто питання з охорони праці;
- створено програмну документацію.

Завдяки розробці програмного забезпечення вебсайту «International» клієнт зможе отримати усю необхідну інформацію про ресторан дистанційно. Також програмне забезпечення вебсайту «International» дає змогу переглянути меню, зареєструватися, авторизуватися, оформити замовлення та забронювати місце на певний час і дату.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Вебсайт «Beer-Altbier». URL: <http://www.beer-altbier.com.ua/> (дата звернення 04.03.2022).
- 2 Вебсайт «Sfood – PapaSHA!». URL: <https://sfood.in.ua/> (дата звернення 04.03.2022).
- 3 Вебсайт «Zelda». URL: <https://zelda.com.ua/> (дата звернення 04.03.2022).
- 4 Мова UML: Діаграма використання. URL: <http://p4ilka.blogspot.com/2018/12/uml.html> (дата звернення 14.03.2022).
- 5 Діаграма діяльності. URL: <https://uk.theastrologypage.com/activity-diagram> (дата звернення 23.03.2022).
- 6 Діаграма «Сутність Зв'язку». URL: <https://studfile.net/preview/5462334/page:12/> (дата звернення 24.03.2022).
- 7 Користувальницький інтерфейс. URL: <http://um.co.ua/8/8-6/8-64888.html> (дата звернення 27.03.2022).
- 8 Стиль інтерфейсу : значення сірого кольору. URL: <https://uk.warbletoncouncil.org/psicologia-significado-color-gris-1307> (29.03.2022).
- 9 Стиль інтерфейсу : значення білого кольору. URL: <https://dovidka.biz.ua/shho-oznachaye-biliy-kolir/> (дата звернення 29.03.2022).
- 10 Стиль інтерфейсу : значення чорного кольору. URL: <https://vabos.com.ua/shho-oznachaye-chornij-kolir-v-psixologi%D1%97/> (дата звернення 29.03.2022).
- 11 Стиль інтерфейсу : значення червоного кольору. URL: <https://bazilik.media/znachennia-koloriv-u-dyzajni-ta-marketynhu/> (29.03.2022).

12 Стиль інтерфейсу : поради щодо використання кольорів у інтерфейсі.
URL: <https://www.adobe.com/ua/creativecloud/design/discover/color-meaning.html> (дата звернення 01.04.2022).

13 Стиль інтерфейсу : сайт шрифтів. URL: <https://fonts.google.com/> (дата звернення 01.04.2022).

14 Діаграма класів. URL: https://www.wiki.uk-ua.nina.az/%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0_%D0%BA%D0%BB%D0%B0%D1%81%D1%96%D0%B2.html (дата звернення 05.04.2022).

15 Діаграма послідовності. URL: <https://studfile.net/preview/9877319/page:2/> (дата звернення 07.04.2022).

16 Логічна модель. URL: https://stud.com.ua/97320/informatika/logichna_model_danih_ponyattya_normalizatsiyi_vidnosin (дата звернення 09.04.2022).

17 VueJs : Інформація про технологію. URL: <https://ru.vuejs.org/v2/guide/index.html> (дата звернення 12.04.2022).

18 JavaScript : Інформація про технологію. URL: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/JavaScript_basics (дата звернення 12.04.2022).

19 HTML : Інформація про технологію. URL: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/HTML_basics (дата звернення 12.04.2022).

20 CSS : Інформація про технологію. URL: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/CSS_basics (дата звернення: 12.04.2022).

21 FireBase: Інформація про технологію. URL: <https://firebase.google.com/> (дата звернення 12.04.2022).

22 Фізична модель даних : термінологія. URL: <https://artsandculture.google.com/entity/m033cp1?hl=uk> (дата звернення 10.04.2022).

23 Фізична модель : проєктування. URL:
https://stud.com.ua/77262/informatika/izicheskie_modelyuvannya (дата звернення 10.04.2022).

24 Діаграма розгортання. URL:
<https://studfile.net/preview/5010027/page:6/> (дата звернення 16.04.2022).

25 Тестування програм, та систем. URL:
https://pidru4niki.com/1628011847733/informatika/testuvannya_program_sistem
(дата звернення 10.05.2022).

26 Про охорону праці : Закон України від 14.10.1992 № 2694-XII. Дата оновлення: 27.02.2021. URL: <https://zakon.rada.gov.ua/laws/show/2694-12#Text> (дата звернення: 15.05.2022).

27 MVC (Model-View-Controller). URL:
<https://www.codecademy.com/article/mvc> (дата звернення 20.05.2022).

ДОДАТОК А ТЕХНІЧНЕ ЗАВДАННЯ

Вступ

Найменування програми – програмне забезпечення вебсайту для ресторану «International».

Коротка характеристика використання:

Програмне забезпечення вебсайту «International» дозволяє клієнту переглянути меню, зареєструватися, авторизуватися, оформити замовлення та забронювати місце на певний час і дату.

1 Підстава для розробки

1.1 Підстави для проведення розробки

Підставою для розробки є завдання з кваліфікаційної роботи.

1.2 Найменування та умовне позначення теми розробки

Найменування розробки – Розробка програмного забезпечення вебсайту для ресторану «International».

2 Призначення розробки

2.1 Функціональне призначення

Функціональним призначенням вебсайту є надання можливості клієнту формування власного замовлення з меню, ознайомлення із рестораном, та можливість зареєструватися, щоб створювати замовлення та бронювати місця.

Для менеджера функціональним призначенням є формування елементу у меню, редагування елементу, редагування замовлення клієнтів.

2.2 Експлуатаційне призначення

Програмне забезпечення призначене для експлуатації на вебсервері.

3 Вимоги до програми або програмного продукту

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до переліку виконуваних функцій

Розроблена програма має надавати можливість виконувати наступні функції:

- Реєстрація;
- Авторизація;
 1. Увійти у систему як Клієнт;
 2. Увійти у систему як Адміністратор.
- Перегляд головної сторінки та меню сайту без авторизації;
 1. Якщо користувач не авторизувався, він має лише можливість оглядати сайт та оглянути меню, а у разі взаємодії із функціоналом – його перенаправить на сторінку реєстрації.
- Можливості взаємодії із сайтом Клієнтом;
 1. Перейти у меню;
 2. Огляд меню;
 3. Додати елемент із меню у кошик замовлення;
 4. Перегляд галереї;
 5. Перехід по посиланнях у меню панелі навігації;
 6. Можливість зайти у свій профіль;
 7. Редагування замовлення і підтвердження його;

8. Отримання інформації про виконане замовлення.
 - Можливості взаємодії із сайтом Менеджером;
1. Перейти у меню;
2. Огляд меню;
3. Додати новий елемент у меню;
4. Редагування елемента у меню;
5. Видалити елемент з меню;
6. Перегляд усіх зареєстрованих користувачів;
7. Перегляд усіх замовлень клієнтів;
8. Редагування замовлення клієнта;
9. Видалення замовлення клієнта.

Вищеописані функції дадуть можливість зручно користуватися вебсайтом. Вся інформація та дані будуть надійно зберігатися у базі.

3.1.2 Вимоги до організації вхідних та вихідних даних

Вхідними даними для Клієнта є:

- 1) Текстова інформація – для введення даних для реєстрації, та авторизації. Буду здійснюватися з клавіатури, або з сенсорного екрана зі смартфона;
- 2) Зображення у форматі .png або .jpg – додавання зображення для зображення на елементу у меню;
- 3) Додавання елементів з меню у власний кошик замовлення.

Вихідні дані для Клієнта:

- 1) Номер замовлення.

Вхідні дані для Менеджера:

- 1) Текстова інформація – для введення даних для реєстрації, та авторизації. Буду здійснюватися з клавіатури, або сенсорного екрану зі смартфона;
- 2) Зображення у форматі .png або .jpg – додавання зображення елементу у меню.

Вихідні дані для Менеджера:

- 1) Текстове зображення інформації на екрані;
- 2) Інформація меню;
- 3) Графічне зображення елементу у меню.

3.1.3 Вимоги до часових обмежень

Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.) потрібно зробити 10.06.2022.

3.2 Вимоги до надійності

3.2.1 Вимоги до забезпечення надійного функціонування програми

Надійне функціонування програми буде лише в тому випадку, якщо дотримуватися виконання організаційно-технічних заходів.

3.2.2 Час відновлення після відмови

Час відновлення після відмови, викликаній несправністю технічних засобів, фатальною помилкою в операційній системі, не повинен перевищувати часу, необхідного на усунення неполаджених технічних засобів і перевстановлення програмних засобів.

3.2.3 Відмови з причини не коректних дій оператора

Встановлення валідації, та передбачення можливої некоректної поведінки користувача. Надати користувачеві інструкцію по експлуатації програми.

3.3 Умови експлуатації

3.3.1 Кліматичні умови експлуатації

Кліматичні умови експлуатації, при яких повинні забезпечуватися задані характеристики, повинні задовольнити вимогам, що пред'являються до технічних засобів у частині умов їх експлуатації.

3.3.2 Вимоги до видів обслуговування

Програма не вимагає проведення будь-яких видів обслуговування.

3.3.3 Вимоги до чисельності та кваліфікації персоналу

Вимоги до чисельності та кваліфікації персоналу не висуваються.

3.4 Вимоги до складу та параметрів технічних засобів

Інтернет і браузер, та власний пристрій: комп'ютер чи смартфон.

3.5 Вимоги до технічної та програмної сумісності

3.5.1 Вимоги до інформаційних структур та методів розв'язку

Для зберігання інформації необхідно використовувати базу даних. Для зберігання даних інформації про клієнта будуть створені такі атрибути:

- 1) ім'я;
- 2) пошта;
- 3) роль;
- 4) пароль.

Для створення елемента з меню потрібно розробити окремі атрибути для трьох категорій:

Категорія блюда:

- 1) Назва блюда;
- 2) Ціна;
- 3) Вага;
- 4) Короткий опис;
- 5) Зображення.

Категорія десертів:

- 1) Назва десерту;
- 2) Ціна;
- 3) Вага;
- 4) Короткий опис;
- 5) Зображення.

Категорія напоїв

- 1) Назва десерту;
- 2) Ціна;
- 3) Вага;
- 4) Короткий опис;
- 5) Зображення.

3.5.4 Вимоги до захисту інформації та програм.

Вимоги до захисту інформації та програм не висуваються.

3.6 Вимоги до маркування та упакування

Доступ до вебсайту буде надаватися через браузер, тому потрібен лише інтернет, та особистий пристрій: комп'ютер чи смартфон.

3.6.1 Вимоги до маркування

Вимоги до маркування не висуваються.

3.6.2 Вимоги до упакування

Вимоги до упакування не висуваються.

3.6.3 Умови пакування

Вимоги до умов пакування не висуваються.

3.6.4 Порядок пакування

Вимоги до порядку пакування не висуваються.

3.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не висуваються.

3.7.1 Умови транспортування та зберігання

Вимоги щодо умов транспортування та зберігання не висуваються.

3.7.2 Спеціальні вимоги

Програма має забезпечувати взаємодію з користувачем за допомогою графічного інтерфейсу користувача, розробленого згідно з побажаннями замовника.

4 Вимоги до програмної документації

4.1 Попередній склад програмної документації

Склад програмної документації має мати (в собі):

- 1) Технічне завдання;
- 2) Програму та методику випробувань;
- 3) Інструкцію користувача;
- 4) Опис програмного забезпечення.

5 Техніко-економічні показники

Орієнтовна економічна ефективність не розраховується.

5.1 Економічні переваги

Програмне забезпечення є ініціативною розробкою, вигода отримується через відсутність виплат заробітної платні.

6 Стадії та етапи розробки програмного забезпечення

6.1 Стадії розробки та етапи розробки Розробка має бути проведена в п'ять етапів:

- 1) Розробка технічного завдання;
- 2) Ескізний проєкт;
- 3) Технічний проєкт;
- 4) Робочий проєкт;
- 5) Впровадження.

6.2 Склад робіт по етапах

На етапі розробки технічного завдання необхідно виконати наступні задачі:

- 1) Аналіз предметної галузі;
- 2) Виявлення та уточнення вимог до технічних засобів;
- 3) Виявлення вимог до програми;
- 4) Виявлення вимог до складу програмної документації. Щодо програмної документації користуватись ДСТУ 19.101-77 «Попередній склад програмної документації».
- 5) Вибір мови програмування та інших програмних засобів;
- 6) Узгодження та затвердження технічного завдання;
- 7) Постановка задачі.

На етапі ескізного проєктування необхідно виконати наступні кроки:

- 1) Змоделювати сценарії роботи програмного забезпечення за допомогою діаграми варіантів використання з наступною розробкою специфікацій варіантів використання.
- 2) Змоделювати сценарії роботи основних варіантів використання за допомогою діаграми діяльності.
- 3) Розробити концептуальну модель даних програмного забезпечення у вигляді діаграми класів.
- 4) Розробити проєкт користувацького інтерфейсу.

На етапі технічного проєктування необхідно виконати наступні види робіт:

- 1) Розробити статичну модель (діаграму класів).

- 2) Створити специфікації змодельованих класів.
- 3) Виконати динамічне моделювання програми за допомогою діаграми послідовностей або діаграми кооперацій для основних варіантів використання.
- 4) Розробити логічну модель даних на основі інформаційної моделі розроблюваного програмного забезпечення.

На етапі робочого проєктування необхідно виконати наступні види робіт:

- 1) Обґрунтувати доцільність використання обраної мови програмування та інших програмних засобів.
- 2) Виконати моделювання компонентів програмного забезпечення за допомогою діаграми компонентів.
- 3) Виконати моделювання розгортання системи за допомогою діаграми розгортання.
- 4) Виконати розробку та тестування програмних модулів.
- 5) Виконати розробку інструкції користувача.
- 6) Провести випробування розробленого програмного забезпечення.

Відповідно випробування розробленого програмного забезпечення включає наступні етапи:

- 1) Розробка, узгодження та затвердження програми та методики випробувань;
- 2) Виправлення програми та програмної документації за результатами випробувань;

3) Проведення приймально-здавальних робіт. На етапі впровадження необхідно виконати роботи з підготовки та передачі розробленого програмного забезпечення та програмної документації Замовнику.

Строки виконання етапів розробки програмного забезпечення можна представлено у таблиці А.1:

Таблиця А.1 – Строки виконання етапів розробки програмного забезпечення вебсайту «International»

Стадії розробки	Назва етапу	Початок роботи	Кінець роботи
1	2	3	4
1 Технічне завдання	Аналіз предметної області	1.02.2022	10.02.2022
	Виявлення вимог до технічних засобів	10.02.2022	28.02.2022
	Виявлення вимог до програми	28.02.2022	10.03.2022
	Виявлення вимог до складу програмної документації	10.03.2022	20.03.2022
	Вибір мови програмування	20.03.2022	24.03.2022
	Узгодження та затвердження технічного завдання	24.03.2022	28.03.2022
	Постановка задачі	28.03.2022	31.03.2022
2 Ескізний проєкт	Проектування поведінки за допомогою варіантів використання	31.03.2022	10.04.2022
	Специфікація варіантів використання	10.04.2022	15.04.2022
	Моделювання діяльності варіантів використання	15.04.2022	20.04.2022
	Розробка концептуальної моделі	20.04.2022	25.04.2022

Продовження таблиці А.1

Стадії розробки	Назва етапу	Початок роботи	Кінець роботи
1	2	3	4
2 Ескізний проєкт	Розробка проєкт користувальницького інтерфейсу	25.04.2022	27.04.2022
3 Технічний проєкт	Розробка статичної моделі	28.04.2022	01.05.2022
	Розробка специфікації статичної моделі	01.05.2022	03.05.2022
	Динамічне моделювання поведінки системи	03.05.2022	06.05.2022
	Логічне моделювання системи на основі інформаційної моделі	06.05.2022	13.05.2022
4 Робочий проєкт	Обґрунтування обраних програмних засобів	13.05.2022	14.05.2022
	Моделювання компонентів системи	14.05.2022	15.05.2022
	Моделювання розгортання системи	15.05.2022	17.05.2022
	Розробка програмних модулів	17.05.2022	18.05.2022
	Тестування програмних модулів	18.05.2022	19.05.2022
	Розробка інструкції користувача	19.05.2022	20.05.2022
	Розробка програми та методики випробувань	20.05.2022	01.06.2022
5 Впровадження	Виправлення програми та програмної документації за результатами випробувань	01.06.2022	05.06.2022
	Проведення приймально-здавальних робіт	06.06.2022	07.06.2022

7 Порядок контролю та приймання

7.1 Види випробувань

Контроль за кожним етапом проектування програмного забезпечення відбувається з урахуванням усіх вимог, визначених в цьому технічному завданні. Кожна стадія розробки повинна бути представлена в зазначені строки та узгоджена із замовником.

Приймально-здавальні роботи мають проводитись під наглядом Замовника або уповноваженої ним особи з 06.06.2022 по 07.06.2022.

Порядок контролю здійснюється відповідно до процедур та методів випробувань і документується.

Приймально-здавальні випробування повинні проводитись не пізніше 07.06.2022 згідно з документом «Програма та методика випробувань», який узгоджується між стороною розробника та стороною замовника.

Хід проведення приймально-здавальних робіт Замовник та Виконавець документують в «Протокол проведення випробувань».

7.2 Загальні вимоги до приймання робіт

На основі Протоколу проведення випробувань Виконавець та Замовник підписують Акт приймання здачі програми в експлуатацію.

ДОДАТОК Б ПРОГРАМА ТА МЕТОДИКА ВИПРОБОВУВАНЬ
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ ДЛЯ РЕСТОРАНУ
«INTERNATIONAL»

1 Об'єкт випробувань

Об'єктом випробування є програмне забезпечення вебсайту для ресторану «International».

2 Мета випробувань

Мета випробування є перевірка працездатності програмного забезпечення, перевірка виконання головних функцій, задач, вимог програмного забезпечення за технічним завданням.

3 Вимоги до вебсайту

Вимогами до вебсайту перевіряються головні вимоги, що були описані у технічному завданні.

4 Вимоги до програмної документації

4.1 Склад програмної документації, що пропонується на випробування
Документація програми включає у себе такі документи як:

- 1) технічне завдання;
- 2) програму та методику випробувань;
- 3) інструкцію користувача;
- 4) опис програмного забезпечення.

5 Технічні вимоги та засоби для користування програмним забезпеченням

Технічні вимоги та пояснення до них є важливою частиною для експлуатації та користуванням із програмних забезпеченням для користувача. Інформація про систему вимог допоможе з'ясувати можливість користуванням програмою до запуску цієї програми. Нижче представлені актуальні системи вимог: браузерів, версій, розміру у пам'яті, мінімальні та рекомендовані вимоги для користувача, та для менеджера.

Загальні вимоги до браузерів представлені у таблиці Б.1.

Таблиця Б.1. Загальні вимоги до версій браузерів.

Браузери	Версія	Розмір
Яндекс	21.11.2.777	2,57 Mb
Firefox	94.0.2	326 Kb
Opera	82.0.4227.23	2,49 Mb
Internet Explorer	11.2273.14393.0	32,8 Mb
Safari	5.1.7	36,7 Mb
Google Chrome	96.0.4664.93	1,27 Mb
Orbitum	56.0.0924.89	48,9 Mb

5.1 Системи вимоги для Клієнта

Системи вимог Персонального Комп'ютеру чи ноутбуку для Клієнту представлено у таблиці Б.2.

Таблиця Б.2. Система вимог для Клієнта

	Мінімальні вимоги	Рекомендовані вимоги
Відеокарта	400 MGb	1 Gb
Процесор	900 MHz	1.4 GHz
Операційна пам'ять	1 GB RAM	1 GB RAM
Монітор	VGA 1280x600	HDMI 1920x1080
Місце у пам'яті на HDD / SSD	Відсутнє	Відсутнє
Оптичний накоплювач	Відсутнє	Відсутнє
Пристрій для взаємодії з користувачем	Клавіатура та комп'ютерна миша	Клавіатура та комп'ютерна миша
Інші пристрої	Гарнітура	Гарнітура
Інтернет	Від 1 Мб/с	Від 3 Мб/с
Операційна система	Windows 7/ Vista/8/ 10 (32/64-розрядна версія). Linux 4.8.0-53-generic (86/64 - розрядна версія). macOS 13.2 (20D64)	Windows 7/ Vista/8/ 10 (32/64-розрядна версія). Linux 4.8.0-53-generic (86/64 - розрядна версія). macOS 13.2 (20D64)

5.2 Система вимог смартфону або планшету для Клієнту представлено у таблиці Б.3.

Таблиця Б.3. Система вимог для Клієнта

	Мінімальні вимоги	Рекомендовані вимоги
Процесор	900 MHz	1.4 GHz
Операційна пам'ять	1 GB RAM	1 GB RAM
Місце у пам'яті на HDD / SSD	89 МБ	89 МБ
Оптичний накоплювач	Відсутнє	Відсутнє
Пристрій для взаємодії з користувачем	Сенсорна	Сенсорна
Інші пристрої	Гарнітура	Гарнітура
Інтернет	Від 1 Мб/с	Від 3 Мб/с

5.3 Системи вимоги для Менеджера

Системні вимоги Персонального Комп'ютеру чи ноутбуку для Менеджера представлено у таблиці Б.4.

Таблиця Б.4. Системні вимоги для Менеджера

	Мінімальні вимоги	Рекомендовані вимоги
Відеокарта	400 MGb	1 Gb
Процесор	900 MHz	1.4 GHz
Операційна пам'ять	1 GB RAM	1 GB RAM
Монітор	VGA 1280x600	HDMI 1920x1080
Місце у пам'яті на HDD / SSD	Відсутнє	Відсутнє
Оптичний накоплювач	Відсутнє	Відсутнє
Пристрій для взаємодії з користувачем	Клавіатура та комп'ютерна миша	Клавіатура та комп'ютерна миша
Інші пристрої	Гарнітура	Гарнітура
Інтернет	Від 1 Мб/с	Від 3 Мб/с
Операційна система	Windows 7/ Vista/8/ 10 (32/64-розрядна версія). Linux 4.8.0-53-generic (86/64 - розрядна версія). macOS 13.2 (20D64)	Windows 7/ Vista/8/ 10 (32/64-розрядна версія). Linux 4.8.0-53-generic (86/64 - розрядна версія). macOS 13.2 (20D64)

6 Методи випробувань

6.1 Методика проведення перевірки вимог до функціональних характеристик програмного забезпечення

Перевірка працездатності програми виконується за пунктом «Вимог до функціональних характеристик» технічного завдання.

Перевірку можна вважати успішною у випадку відповідності виконуваних функцій згідно за технічним завданням.

Мета випробування є функціональність, працездатність вебсайту «International». Перелік випробувань представлено у таблиці Б.5:

Таблиця Б.5 – Перелік випробувань

№	Дія	Очікуваний результат
1	2	3
1	Відкрити програмне забезпечення у браузері	Відображення головної сторінки
2	Натиснути на надпис «LogIn»	Перейти на сторінку авторизації
3	Натиснути «Зареєструватися» на сторінці з авторизацією	Перейти на сторінку реєстрації
4	Ввести не коректні дані у панель	Отримати коментарій валідації
5	Натиснути кнопку «Увійти» та не вводити ніяких даних у панель	Отримати коментарій валідації
6	У меню навігаційної панелі натиснути: Меню, Галерея, Опис, Контакти, тощо	Переміщуватися по сторінці сайту через внутрішні посилання.
7	Натиснути кнопку «Меню»	Перейти на сторінку Меню
8	Натиснути на додати елемент у замовлення	Отримати цей елемент у замовлення
9	Підтвердити замовлення	Отримати коментарій повідомлення про виконану операцію
10	Редагування свого замовлення	Отримати нові дані замовлення

Продовження таблиці Б.5

№	Дія	Очікуваний результат
11	Додати новий елемент у Меню	Отримати новий елемент у Базі Даних
12	Редагувати елемент з Меню	Отримати нові дані елементу з Меню
13	Натиснути на надпис «Профіль»	Перейти на сторінку свого профілю
14	Редагування свого профілю	Отримати нові дані профілю
15	Взаємодіяти із сайтом не авторизувавшись	Перехід до сторінки авторизації

ДОДАТОК В ІНСТРУКЦІЯ КОРИСТУВАЧА ДЛЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ ДЛЯ РЕСТОРАНУ «INTERNATIONAL»

Представлено наступну інструкцію разом із рисунками, щоб наглядно показати роботу вебсайту.

Крок 1. Ви перейшли на головний вебсайту ресторану «International». Вас зустрічає головне вікно, навігаційне меню (угорі), та надпис сайту. Ви зайшли як «Гість». Гість може вільно пересуватися по головному сторінці сайту. У навігаційному меню є такі атрибути як: «Інформація про сайт або About», «Меню», «Галерея», «Новини», «Контакти», завдяки яким можна переміщуватися, або «пригати» по заголовкам сайту, завдяки внутрішнім посиланням, це спростить огляд сайту.

Результат вхід на сторінку сайту, та огляд самого сайту показано на рисунках В.1 – В.6.



Рисунок В.1 – Вхід до головної сторінки вебсайту «International»

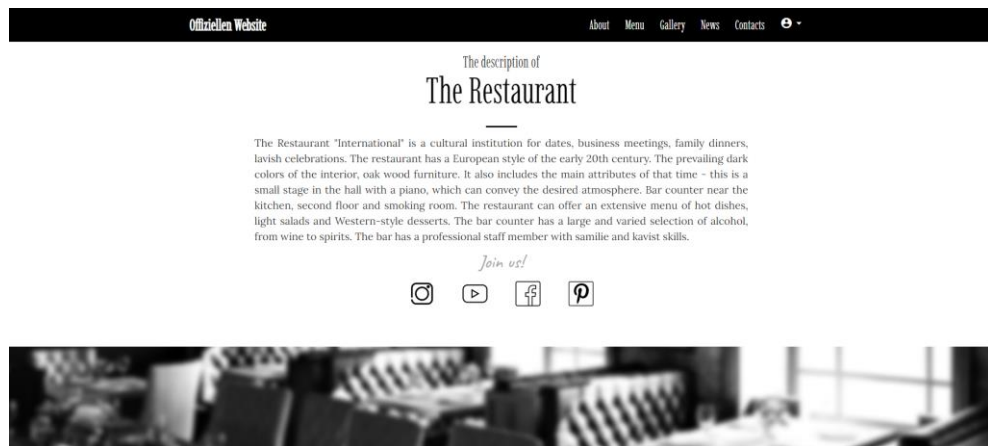


Рисунок В.2 – Інформація про сайт

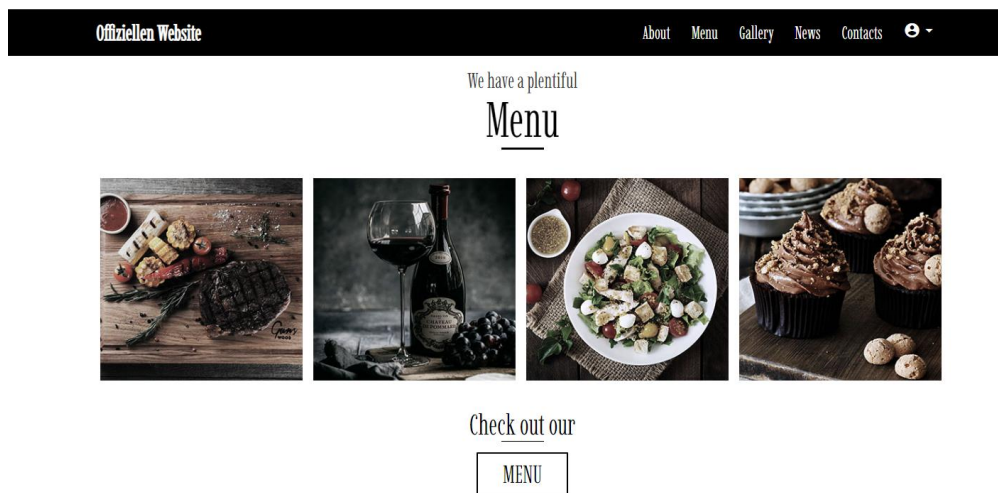


Рисунок В.3 – Меню сайту

Look at our

Gallery



Рисунок В.4 – Галерея сайту

What's
New?

Would you like some coffee or tea?



As in any restaurant, we are always ready to offer you a variety of drinks! We have professional staff at the bar, you can order a variety of coffees while waiting, for romantic dates we have a large assortment of wines, shapan, martinis and other types, for a friendly company we have a proud selection of beer, cider, ale and other fruit drinks. We always have what we can surprise and delight you with!

What could we offer you?

We also have an extensive selection of dishes, desserts, light salads. Our cuisine is specialized in the style of European cuisine. Our team of chefs can surprise you with their cooking skills. Our dishes will accompany you from Italy to Scandinavia, from France to Eastern Europe! We believe it is better to specialize in one culture than to try to cover all cultures of international cuisines, but we are happy to feed all visitors from all over the world with our dishes!



Рисунок В.5 – Новини сайту

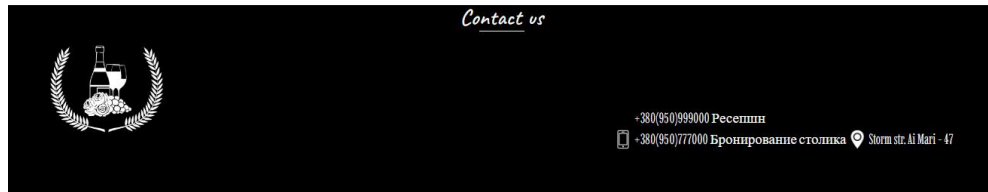


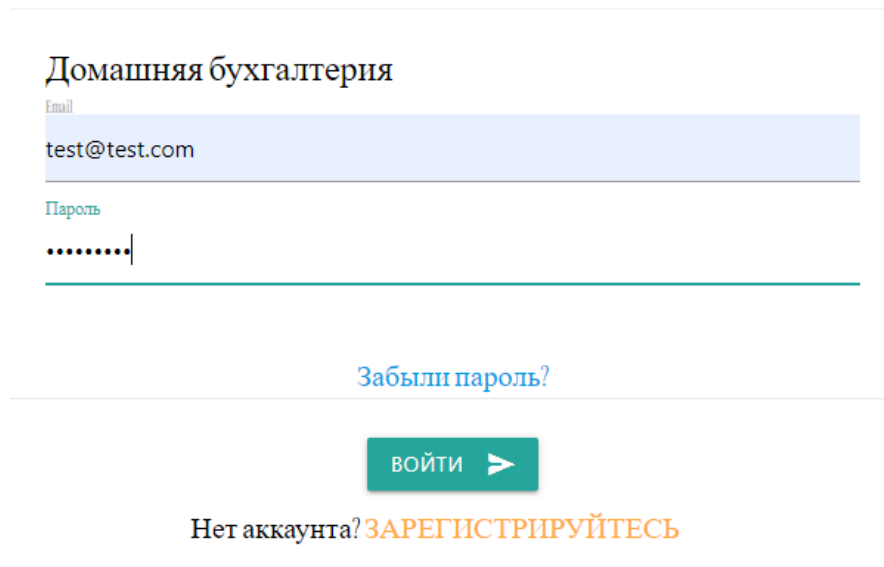
Рисунок В.6 – Контакти сайту

Крок 2. Щоб користуватися сайтом повноцінно як користувач, треба зареєструватися, та авторизуватися у систему.

Реєстрація користувача наведена на рисунку В.7.

Рисунок В.7 – Реєстрація користувача

Результат авторизація користувача наведена на рисунку В.8.



Домашня бухгалтерия

Email

test@test.com

Пароль

.....|

[Забыли пароль?](#)

[ВОЙТИ >](#)

Нет аккаунта? [ЗАРЕГИСТРИРУЙТЕСЬ](#)

Рисунок В.8 – Авторизація користувача

Результат входу у систему користувачем наведена на рисунку В.9.



Рисунок В.9 – Результат входу у систему

Крок 3. Тепер користувач увійшов у систему, наглядно показано на рисунку В.9, де у навігаційному меню, при натисканні у іконку профілю, відкривається вікно, де ми можемо перейти у Профіль, до замовлення, або вийти із системи.

Далі, ми зайдемо у Меню, щоб створити сет замовлення. Результат входу у Меню наведена на рисунку В.10.

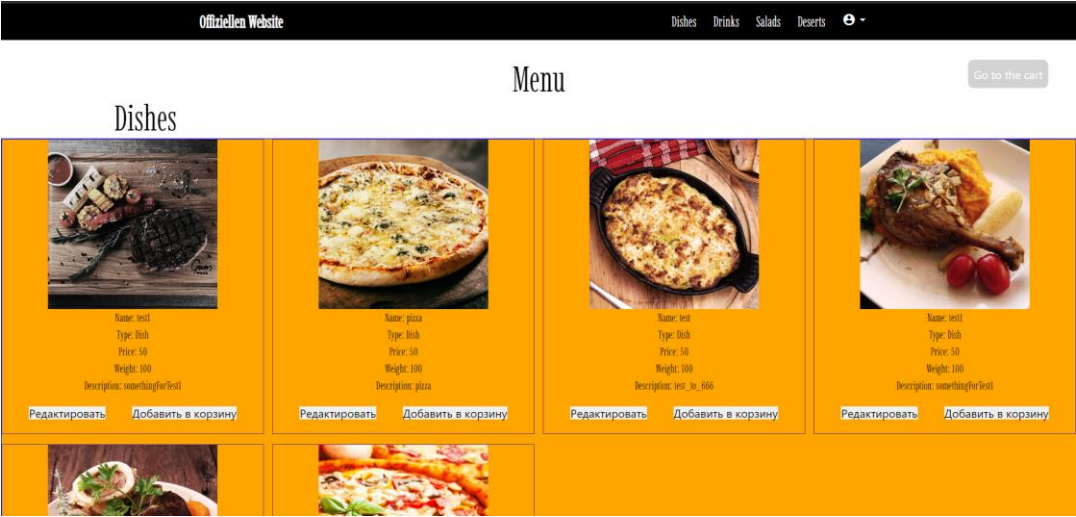


Рисунок В.10 – Вхід до Меню

Перед тим як додавати елемент у корзину, подивимось що у нас лежить у самій корзині. Перевірка Корзини до додавання товару наведена у рисунку В.11.

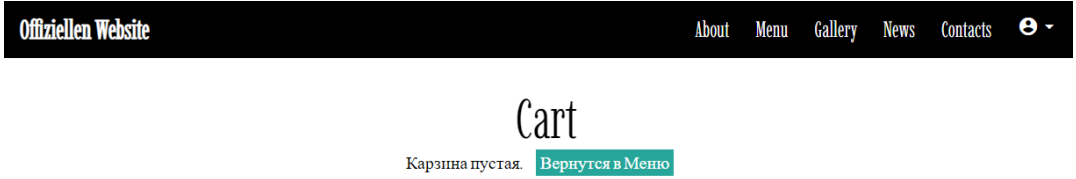


Рисунок В.11 – Перевірка корзини до додавання товару у Корзину

Як бачимо, кошик замовлення пуста, про що нам і повідомляє сторінка. Тепер, перевіримо наше замовлення. Перевірка Замовлення до підтвердження наведена на рисунку В.12.



Рисунок В.12 – Перевірка пустого Кошика

Перегляд корзини після додавання декількох елементів у Корзину наведено на рисунку В.13.

Cart

#	Название	Тип	Вес	Цена	Убрать
1	Неолитанская пицца	Dish	100	50	
2	Утка в меду	Dish	190	210	
3	Котлета по Киевски	Dish	150	90	

☐ Reservation a table

[Подтвердить заказ](#) [Отменить заказ](#)

Рисунок В.13 – Огляд замовлення клієнта з елементами

Ми можемо забронювати стіл на певний час й дату. Огляд кошик замовлення з бронюванням столу наведено на рисунку В.14, В.15.

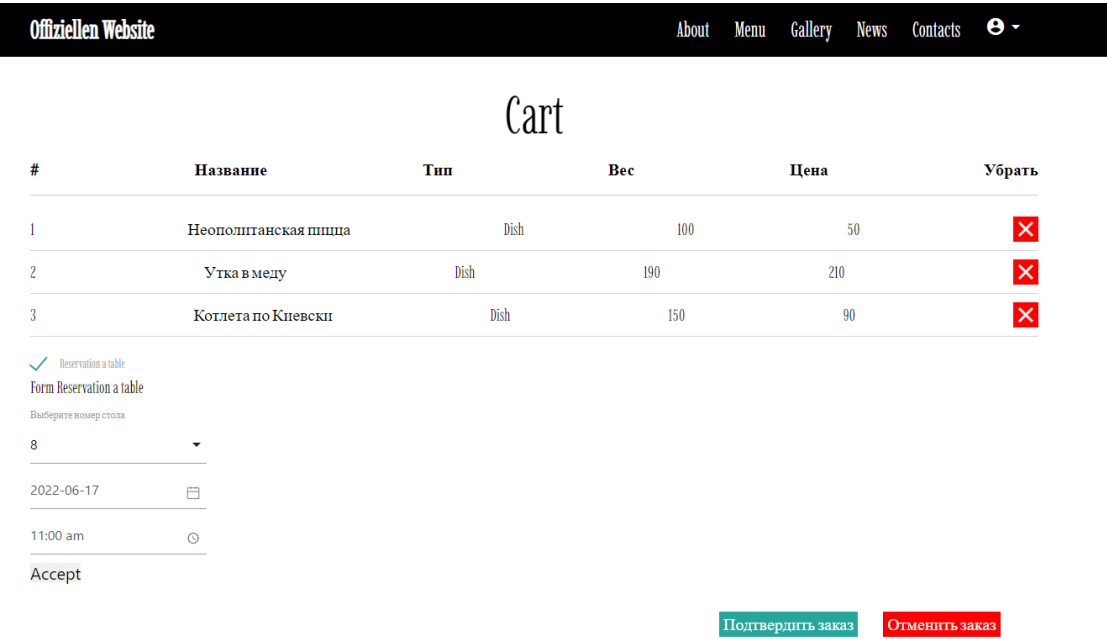


Рисунок В.14 – Огляд кошик замовлення з бронюванням столу

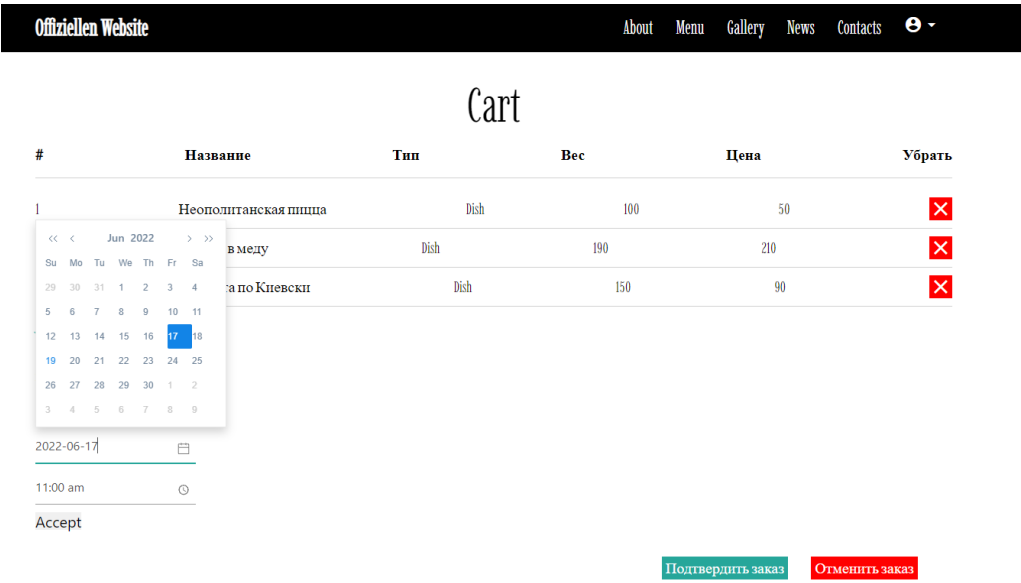


Рисунок В.15 – Огляд кошик замовлення з бронюванням столу

Крок 4. Після того, як ми закінчили набирати сет, можна його підтвердити. Дані про ваше замовлення буде у вікні «Моє замовлення» чи «My Orders». Оглянемо наше замовлення у першому випадку (без бронювання столу), та у другому (із бронюванням столу). Перегляд Замовлення після підтвердження замовлення у Корзині наведено на рисунку В.16.

Offiziellen Website		About	Menu	Gallery	News	Contacts	⊕
		Order					
		Заказ №61807					
Наименование:	Лукавый суп.Котлета по Киевски						
Стоимость:	280,90						
Итоговая цена:	370 ₤						
Дата:	19.06.2022						
Номер столика:	-						
Дата бронирование:	-						
Время бронирование:	-						
		Редактировать ✕					
		Заказ №70919					
Наименование:	Котлета по Киевски,Утка в меду.Котлета по Киевски						
Стоимость:	90,210,90						
Итоговая цена:	390 ₤						
Дата:	19.06.2022						
Номер столика:	6						
Дата бронирование:	2022-06-17						
Время бронирование:	10:00						
		Редактировать ✕					

Рисунок В.16 – Огляд списку Замовлення

Від імені адміністратора можна додати елемент у Меню, маніпулювати із Базою Даних, та інше. Наведемо приклад додання елементу у Меню. Сторінка вебсайту Адміністратора наведено на рисунку В.17.



Рисунок В.17 – Сторінка Адміністратора

Сторінка Адміністратора додавання елементу у Меню наведено на рисунку В.18.

Offiziellen Website

Admin

TEST

Current User

Name: Admin

Role: admin

1

New Element Menu

Выберите категорию

Dish

Название

test1

Вес

100

Цена

50

Описание

somethingForTest1

Загрузить

Изображение

СОЗДАТЬ

Рисунок В.18 – Додавання елементу у Меню

ДОДАТОК Г КОД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ ДЛЯ РЕСТОРАНУ «INTERNATIONAL»

Код index.html

```
<!DOCTYPE html>
<html lang="">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <link rel="icon" href="<%= BASE_URL %>favicon.ico">
    <!-- Для адаптивной верстки-->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="assets/css/style.css">
    <!-- Imbue font: for main title -->
    <!-- font-family: 'Imbue', serif; -->
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
href="https://fonts.googleapis.com/css2?family=Imbue:wght@300&display=swap"
rel="stylesheet">
    <!-- Bebas Neue font -->
    <!-- font-family: 'Bebas Neue', cursive; -->
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
href="https://fonts.googleapis.com/css2?family=Bebas+Neue&display=swap"
rel="stylesheet">
    <!-- Cinzel font: Semi-bold -->
    <!-- font-family: 'Cinzel', serif; -->
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
href="https://fonts.googleapis.com/css2?family=Cinzel:wght@600&display=swap"
rel="stylesheet">
    <!-- Fredericka the Great: Chalking style -->
    <!-- font-family: 'Fredericka the Great', cursive; -->
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
href="https://fonts.googleapis.com/css2?family=Fredericka+the+Great&display=s
wap" rel="stylesheet">
    <!-- Lora: Desription typping style -->
```

```

    <!-- font-family: 'Lora', serif; -->
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css2?family=Lora&display=swap"
rel="stylesheet">
    <!-- Caveat -->
    <!-- font-family: 'Caveat', cursive; -->
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
href="https://fonts.googleapis.com/css2?family=Caveat:wght@500&display=swap"
rel="stylesheet">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">

    <title><%= htmlWebpackPlugin.options.title %></title>
</head>
<body>
    <noscript>
        <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't
work properly without JavaScript enabled. Please enable it to
continue.</strong>
    </noscript>
    <div id="app"></div>
    <!-- built files will be auto injected -->
</body>
</html>

```

Код App.vue

```

<template>
  <div id="app">
    <component :is="layout">
      <router-view/>
    </component>
  </div>
</template>

<script>
import EmptyLayout from './layouts/EmptyLayout'
import MainLayout from './layouts/MainLayout'
import AdminLayout from './layouts/AdminLayout'
export default {
  computed: {
    layout() {

```

```

        return (this.$route.meta.layout || 'empty') + '-layout'
    }
},
components: {
    EmptyLayout,
    MainLayout,
    AdminLayout
}
}
</script>

<style>
@import '~materialize-css/dist/css/materialize.min.css';
@import './assets/index.css';
</style>

```

Код Login.vue

```

<template>
  <form class="card auth-card" @submit.prevent="submitHandler">
    <div class="card-content">
      <span class="card-title">Домашняя бухгалтерия</span>
      <div class="input-field">
        <input
          id="email"
          type="text"
          v-model.trim="email"
          :class="{invalid: ($v.email.$dirty && !$v.email.required) ||
($v.email.$dirty && !$v.email.email)}"
        >
        <label for="email">Email</label>
        <small
          class="helper-text invalid"
          v-if="$v.email.$dirty && !$v.email.required"
        >Поле не должно быть пустым</small>
        <small
          class="helper-text invalid"
          v-else-if="$v.email.$dirty && !$v.email.email"
        >Введите корректный Email</small>
      </div>
      <div class="input-field">
        <input
          id="password"

```

```

        type="password"
        v-model.trim="password"
        :class="{invalid: ($v.password.$dirty && !$v.password.required)
|| ($v.password.$dirty && !$v.password.minLength)}"
    >
    <label for="password">Пароль</label>
    <small
        class="helper-text invalid"
        v-if="$v.password.$dirty && !$v.password.required"
    >Введите пароль</small>
    <small
        class="helper-text invalid"
        v-else-if="$v.password.$dirty && !$v.password.minLength"
    >Пароль должен иметь {{$v.password.$params.minLength.min}} символов.
    Сейчас он {{$password.length}}
    </small>
    </div>
</div>
<p class="center">
    <router-link to="/resetPassword">Забыли пароль?</router-link>
</p>
<div class="card-action">
    <div class="btn-card">
        <button
            class="btn waves-effect waves-light auth-submit"
            type="submit"
        >
            Войти
            <i class="material-icons right">send</i>
        </button>
    </div>

    <p class="center">
        Нет аккаунта?
        <router-link to="/register">Зарегистрируйтесь</router-link>
    </p>
    </div>
</form>
</template>

<script>
import {email, required, minLength} from 'vuelidate/lib/validators'

```

```

import messages from "../utils/messages";

export default {
  name: 'login',
  data: () => ({
    email: '',
    password: ''
  }),
  validations: {
    email: {email, required},
    password: {required, minLength: minLength(6)}
  },
  mounted() {
    if (messages[this.$route.query.message]) {
      this.$message(messages[this.$route.query.message])
    }
  },
  methods: {
    async submitHandler() {
      if (this.$v.$invalid) {
        this.$v.$touch()
        return
      }
      const formData = {
        email: this.email,
        password: this.password
      }

      try {
        await this.$store.dispatch('login', formData)
        await this.$router.push('/')
      } catch (e) {
        console.log(e)
      }
    }
  }
}
</script>

<style>
.card {
  width: 100%;

```

```

max-width: 600px;
display: block;
margin-left: auto;
margin-right: auto;
}
.btn-card {
padding: 10px 0;
}
.btn {
display: block;
margin-left: auto;
margin-right: auto;
}
</style>

```

Код Register.vue

```

<template>
  <form class="card auth-card" @submit.prevent="submitHandler">
    <div class="card-content">
      <span class="card-title">Домашняя бухгалтерия</span>
      <div class="input-field">
        <input
          id="email"
          type="text"
          v-model.trim="email"
          :class="{invalid: ($v.email.$dirty && !$v.email.required) ||
($v.email.$dirty && !$v.email.email)}"
        >
        <label for="email">Email</label>
        <small
          class="helper-text invalid"
          v-if="$v.email.$dirty && !$v.email.required"
        >Поле не должно быть пустым</small>
        <small
          class="helper-text invalid"
          v-else-if="$v.email.$dirty && !$v.email.email"
        >Введите корректный Email</small>
      </div>
      <div class="input-field">
        <input
          id="password"
          type="password"

```



```

        v-model.trim="password"
        :class="{invalid: ($v.password.$dirty && !$v.password.required)
|| ($v.password.$dirty && !$v.password.minLength)}"
    >
    <label for="password">Пароль</label>
    <small
        class="helper-text invalid"
        v-if="$v.password.$dirty && !$v.password.required"
    >Введите пароль</small>
    <small
        class="helper-text invalid"
        v-else-if="$v.password.$dirty && !$v.password.minLength"
    >Пароль должен иметь {{ $v.password.$params.minLength.min }} символов.
    Сейчас он {{ password.length }}
    </small>
</div>
<div class="input-field">
    <input
        id="name"
        type="text"
        v-model.trim="name"
        :class="{invalid: $v.name.$dirty && !$v.name.required}"
    >
    <label for="name">Имя</label>
    <small
        class="helper-text invalid"
        v-if="$v.name.$dirty && !$v.name.required"
    >
        Введите ваше имя!
    </small>
</div>
<p>
    <label>
        <input type="checkbox" v-model="agree"/>
        <span>С правилами согласен</span>
    </label>
</p>
</div>
<div class="card-action">
    <div class="btn-card">
        <button
            class="btn waves-effect waves-light auth-submit"

```

```

        type="submit"
      >
        Зарегистрироваться
        <i class="material-icons right">send</i>
      </button>
    </div>

    <p class="center">
      Уже есть аккаунт?
      <router-link to="/login">Войти!</router-link>
    </p>
  </div>
</form>
</template>

<script>
import {email, minLength, required} from "vuelidate/lib/validators";

export default {
  name: 'register',
  data: ()=> ({
    email: '',
    password: '',
    name: '',
    agree: false,
  }),
  validations: {
    email: {email, required},
    password: {required, minLength: minLength(6)},
    name: {required},
    agree: {checked: v => v}
  },
  methods: {
    async submitHandler() {
      if (this.$v.$invalid) {
        this.$v.$touch()
        return
      }

      const formData = {
        email: this.email,
        password: this.password,

```

```

        name: this.name,
    }

    try {
        await this.$store.dispatch('register', formData)
        await this.$router.push('/')
    } catch (e) {
        console.log(e);
    }
}
}
}
</script>

```

Код route.js

```

import Vue from 'vue'
import VueRouter from 'vue-router'
import firebase from "firebase/compat/app";
import Home from '../views/Home'
import Login from '../views/Login'
import Register from '../views/Register'
import ResetPassword from "../views/ResetPassword";
import Profile from '../views/Profile'
import Order from '../views/Order'
import Menu from '../views/Menu'
import AdminHome from '../adminFile/components/Home'
import MenuList from "../adminFile/components/MenuList";
import OrdersHistory from "../adminFile/components/OrdersHistory";
import UsersList from "../adminFile/components/UsersList";
import AddDeserts from "../adminFile/components/Menu/AddDeserts";
import AddDrinks from "../adminFile/components/Menu/AddDrinks";
import AddDishes from "../adminFile/components/Menu/AddDishes";
import AddSalads from "../adminFile/components/Menu/AddSalads";
import Test from '../components/Test'

Vue.use(VueRouter)

const routes = [
  {
    path: '/',
    name: 'Home',
    meta: {layout: 'main', auth: false},

```

```

        component: Home
    },
    {
        path: '/login',
        name: 'login',
        meta: {layout: 'empty'},
        component: Login
    },
    {
        path: '/register',
        name: 'register',
        meta: {layout: 'empty'},
        component: Register
    },
    {
        path: '/resetPassword',
        name: 'resetPassword',
        meta: {layout: 'empty'},
        component: ResetPassword
    },
    {
        path: '/menu',
        name: 'Menu',
        meta: {layout: 'main', auth: true},
        component: Menu
    },
    {
        path: '/order',
        name: 'Order',
        meta: {layout: 'main', auth: true},
        component: Order
    },
    {
        path: '/profile',
        name: 'Profile',
        meta: {layout: 'main', auth: true},
        component: Profile
    },
    {
        path: '/adminHome',
        name: 'AdminHome',
        meta: {layout: 'admin', auth: true},

```

```

        component: AdminHome
    },
    {
        path: '/menuList',
        name: 'MenuList',
        meta: {layout: 'admin', auth: true},
        component: MenuList
    },
    {
        path: '/orderHistory',
        name: 'OrdersHistory',
        meta: {layout: 'admin', auth: true},
        component: OrdersHistory
    },
    {
        path: '/usersList',
        name: 'UsersList',
        meta: {layout: 'admin', auth: true},
        component: UsersList
    },
    {
        path: '/addDishes',
        name: 'AddDishes',
        meta: {layout: 'admin', auth: true},
        component: AddDishes
    },
    {
        path: '/addDrinks',
        name: 'AddDrinks',
        meta: {layout: 'admin', auth: true},
        component: AddDrinks
    },
    {
        path: '/addSalads',
        name: 'AddSalads',
        meta: {layout: 'admin', auth: true},
        component: AddSalads
    },
    {
        path: '/addDeserts',
        name: 'AddDeserts',
        meta: {layout: 'admin', auth: true},

```

```

        component: AddDeserts
      },
      {
        path: '/test',
        name: 'Test',
        component: Test
      }
    ]

    const router = new VueRouter({
      mode: 'history',
      base: process.env.BASE_URL,
      routes

    })

    router.beforeEach((to, from, next) => {
      const currentUser = firebase.auth().currentUser
      const requireAuth = to.matched.some(record => record.meta.auth)

      if (requireAuth && !currentUser) {
        next('/login?message=login')
      } else {
        next()
      }
    })

    export default router

```

Код MainLayout.vue

```

<template>
  <div>
    <div class="app-main-layout">
      <NavbarHelper v-if="getCurrentRoute() === 1 || 2 || 3"/>
      <Navbar v-else/>
      <main>
        <div>
          <router-view :key="$route.path"/>
        </div>
      </main>
    </div>
  </div>
</template>

```

```

<script>
import Navbar from '../components/app/Navbar'
import NavbarHelper from '../components/app/NavbarHelper'
import messages from "../utils/messages";

export default {
  name: 'main-layout',
  data: () => ({
    loading: true
  }),
  components: {
    Navbar, NavbarHelper
  },
  methods: {
    getCurrentRoute() {
      let route = this.$route.path
      console.log(route);
      if(route === '/profile') {
        return 1
      } else if (route === '/menu') {
        return 2
      } else if (route === '/order') {
        return 3
      }
    }
  },
  watch: {
    error(fbError) {
      this.$error(messages[fbError.code] || 'Что-то пошло не так')
    }
  }
}
</script>

```

Код EmptyLayout.vue

```

<template>
  <div class="grey darken-1 empty-layout">
    <router-view />
  </div>
</template>

```

```

<script>
import messages from "../utils/messages";

export default {
  computed: {
    error() {
      return this.$store.getters.error
    }
  },
  watch: {
    error(fbError) {
      this.$error(messages[fbError.code] || 'Что-то пошло не так')
    }
  }
}
</script>

```

Код auth.js

```

import firebase from "firebase/compat/app";

export default {
  actions: {
    async login({commit}, {email, password}) {
      try {
        await firebase.auth().signInWithEmailAndPassword(email,
password)
      } catch (e) {
        commit('setError', e)
        throw e;
      }
    },
    async register({dispatch, commit}, {email, password, name}) {
      try {
        await firebase.auth().createUserWithEmailAndPassword(email,
password)

        const uid = await dispatch('getUid')
        await firebase.database().ref(`/users/${uid}/info`).set({
          name,
          roles: 'user'
        })
      } catch (e) {
        commit('setError', e)
      }
    }
  }
}

```



```

        console.log(commit('setError', e));
        throw e;
    }
},
getUid() {
    const user = firebase.auth().currentUser
    return user ? user.uid : null
},
async logout({commit}) {
    await firebase.auth().signOut()
    await commit('clearInfo')
}
}
}

```

Код indexAuth.js

```

import Vue from 'vue'
import Vuex from 'vuex'
import auth from './auth'
import info from './info'

Vue.use(Vuex)

export default new Vuex.Store({
  state: {
    error: null
  },
  mutations: {
    setError(state, error) {
      state.error = error
    },
    clearError(state) {
      state.error = null
    }
  },
  getters: {
    error: s => s.error
  },
  modules: {
    auth,
    info,
  }
}

```

```
}
```

Код Navbar.vue

```
<template>
  <header class="header header--fixed" id="header">
    <Loader v-if="loading"/>

    <div class="container" v-else>
      <div class="header_inner">
        <a class="header_logo" href="#intro">Offiziellen Website</a>
        <ul class="nav-ul-flex">
          <li>
            <div class="nav">
              <a class="nav_link" href="#link_about">About</a>
              <a class="nav_link" href="#link_menu">Menu</a>
              <a class="nav_link" href="#link_gallery">Gallery</a>
              <a class="nav_link" href="#link_news">News</a>
              <a class="nav_link" href="#link_cont">Contacts</a>
            </div>
          </li>
          <!-- IF USER unlogin-->
          <li v-if="getUser()===0">
            <div class="nav-log">
              <a class="nav_link" href="#" @click.prevent="login">
                LogIn
              </a>
            </div>
          </li>
          <!-- IF USER login -->
          <li class="navbar-dropdown" v-else-if="getUser()===1">
            <!--
              <div class="nav-log">-->
            <!--
              <a class="nav_link" href="#" @click.prevent="logout">-->
            <!--
              LogOut-->
            <!--
              </a>-->
            <!--
              </div>-->

            <div class="nav-log">
              <a class='dropdown-trigger'
                data-target='dropdown'
                ref="dropdown"
              >
```

```

        <i class="material-icons white-text">account_circle</i><i
class="material-icons white-text">arrow_drop_down</i>
      </a>
      <ul id='dropdown' class='dropdown-content'>
        <li><router-link :to="'/profile'" class="navbar-drop-link
black-text">
          <i class="material-icons black-
text">account_circle</i>Profile
        </router-link></li>
        <li class="divider" tabindex="-1"></li>
        <li><router-link :to="'/order'" class="navbar-drop-link
black-text">
          <i class="material-icons black-
text">format_list_bulleted</i>My Orders
        </router-link></li>
        <li class="divider" tabindex="-1"></li>
        <li>
          <a class="navbar-drop-link black-text"
@click.prevent="logout">
            <i class="material-icons black-
text">exit_to_app</i>LogOut
          </a>
        </li>
      </ul>
    </div>
  </li>
</ul>
</div>
</div>
</header>
</template>

<script>
import M from 'materialize-css'
import firebase from "firebase/compat/app";

export default {
  name: "Navbar",
  data: () => ({
    interval: null,
    dropdown: null,
    loading: false

```

```

   )),
  methods: {
    async logout() {
      await this.$store.dispatch('logout')
      this.loading = true
      await this.$router.push('/login?message=logout')
      await this.$router.push('./')
      this.loading = false
    },
    async login() {
      await this.$router.push('/login?message=login')
    },
    getUser() {
      const user = firebase.auth().currentUser
      if (user) {
        return 1
      } else {
        return 0
      }
    }
  },
  computed: {
    name() {
      return this.$store.getters.info.name
    }
  },
  mounted() {
    this.interval = setInterval(()=>{
      this.date = new Date()
    }, 1000)
    this.dropdown = M.Dropdown.init(this.$refs.dropdown, {
      constrainWidth: false,
    })
  },
  beforeDestroy() {
    clearInterval(this.interval)
    if (this.dropdown && this.dropdown.destroy) {
      this.dropdown.destroy()
    }
  }
}
</script>

```

```

<style scoped>
.navbar-dropdown {
  margin: 0 15px;
}
.navbar-drop-link {
  font-size: 20px
}
</style>

```

Код AdminNavbar.vue

```

<template>
  <header class="header header--fixed" id="header">
    <div class="container">
      <div class="header_inner">
        <router-link class="header_logo" to=".">
          Offiziellen Website
        </router-link>
        <ul class="nav-ul-flex">
          <li class="navbar-dropdown">
            <div class="nav-log">
              <a class='dropdown-trigger'
                data-target='dropdown'
                ref="dropdown"
              >
                {{name}}
                <i class="material-icons white-text">account_circle</i><i
class="material-icons white-text">arrow_drop_down</i>
              </a>
              <ul id='dropdown' class='dropdown-content'>
                <li><router-link :to="'/profile'" class="navbar-drop-link
black-text">
                  <i class="material-icons black-
text">account_circle</i>Profile
                </router-link></li>
                <li class="divider" tabindex="-1"></li>
                <li><router-link :to="'/order'" class="navbar-drop-link
black-text">
                  <i class="material-icons black-
text">format_list_bulleted</i>My Orders
                </router-link></li>
                <li class="divider" tabindex="-1"></li>

```

```

        <li>
          <a class="navbar-drop-link black-text"
@click.prevent="logout">
            <i class="material-icons black-
text">exit_to_app</i>LogOut
          </a>
        </li>
      </ul>
    </div>
  </li>
</ul>
</div>
</div>
</header>
</template>

```

```

<script>
import M from 'materialize-css'

export default {
  name: "Navbar",
  data: () => ({
    interval: null,
    dropdown: null,
  }),
  methods: {
    async logout() {
      await this.$store.dispatch('logout')
      // await this.$router.push('./')
      // await this.$forceUpdate()
      // console.log(this.$forceUpdate());
      await this.$router.push('/login?message=logout')
    },
  },
  computed: {
    name() {
      return this.$store.getters.info.name
    }
  },
  mounted() {
    this.interval = setInterval(()=>{
      this.date = new Date()

```

```

    }, 1000)
    this.dropdown = M.Dropdown.init(this.$refs.dropdown, {
      constrainWidth: false,
    })
  },
  beforeDestroy() {
    clearInterval(this.interval)
    if (this.dropdown && this.dropdown.destroy) {
      this.dropdown.destroy()
    }
  }
}
</script>

<style scoped>
.navbar-dropdown {
  margin: 0 15px;
}
.navbar-drop-link {
  font-size: 20px
}
</style>

```

ДОДАТОК Д ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ ДЛЯ РЕСТОРАНУ «INTERNATIONAL»

1 Загальні відомості про програмне забезпечення

1.1 Найменування програми

Найменування: Програмне забезпечення вебсайту для ресторану «International».

1.2 Програмне забезпечення, необхідне для автоматизації процесу формування замовлення.

Для забезпечення належної роботи вебсайту необхідно мати інтернет від 1 Мб/с, та сучасний браузер (GoogleChrome, FireFox, Opera, Yandex та інші), версії яких було представлено у Додатку В у таблиці В.1.

1.3 Мова програмування

Для реалізації програмного забезпечення мною була задіяна мова JavaScript, для реалізації складних задач, було задіяно фреймворк Vue.js, а для зберігання інформації було застосовано СКБД FireBase.

2 Функціональне призначення програмного забезпечення

Функціональне призначення програмного забезпечення – автоматизація процесу формування замовлення, зберігання інформації про замовлення, користувачів, меню, та поліпшення роботи для ресторану.

Функції які доступні користувачу як Клієнту:

- реєстрація;
- авторизація;
- огляд сайту, та меню;
- додавати елементи з меню у замовлення;

- формування замовлення;
- редагування замовлення;
- підтвердження замовлення.

Функції які доступні користувачу як Менеджеру:

- авторизація;
- огляд сайту, та меню;
- додавання елементу у меню;
- редагування елементів у меню;
- огляд усіх зареєстрованих користувачів;
- огляд усіх замовлень клієнтів;
- редагування замовлення клієнта.

2.2 Функціональні обмеження

Щоб робота програмного забезпечення було комфортною, логічною, та зрозумілою, було надано такі обмеження:

- користувач може не реєструватися, але він буде обмежений із роботою сайтом;

1. не може формувати замовлення;
 2. не може переходити у профіль.
- користувач із роллю «Клієнт» обмежений у правах;
1. не може додавати елемент у меню;
 2. не може оглядати усі замовлення клієнтів;
 3. не може редагувати замовлення іншого клієнта;
 4. не може редагувати елемент з меню;
 5. не може видалити елемент з меню;
 6. не може видалити замовлення іншого клієнта;

3 Опис логічної структури

3.1 Структура програми

Програмне забезпечення розроблюється під модель шаблону MVC (Model-View-Controller, Модель-Представлення-Контролер) [27]. Модель шаблону поділяється на три частини:

- 1) Модель – є представлення даних, які обробляє програма;
- 2) Представлення – є елементом, який відповідає за відображення даних моделі користувача.
- 3) Контролер – є елементом, що реагує на дії користувача, та замінює дані у моделі.

Модель MVC наведено на рисунку Д.1:

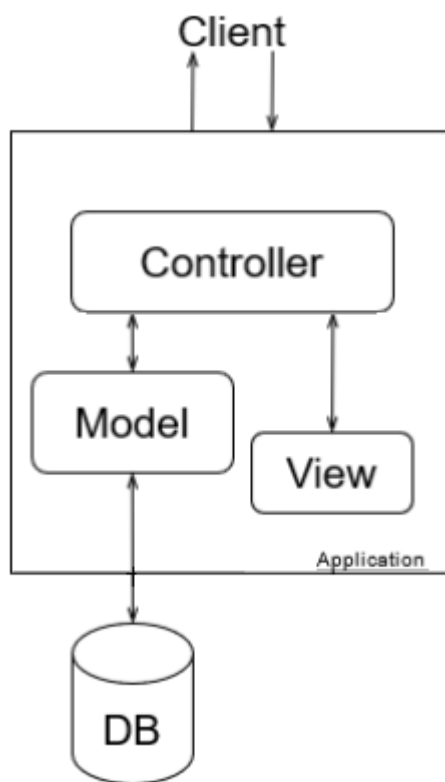


Рисунок Д.1 – Модель MVC

3.2 Алгоритм програми

Алгоритм програми наступний: реєстрація, авторизація, увійти до меню, додати елемент з меню у замовлення, підтвердити замовлення, отримати звіт з замовлення.

3.3 Використовувані методи

Для розробки програмного забезпечення вебсайту «International» було використано мову JavaScript, Vue.js. Для роботи з базами даних було використано метод обробка та зберігання даних.

4 Технічні засоби, що використовуються

Програму забезпечення буде виконуватися на особистих комп'ютерних пристроях, або смартфонах. Вимоги до персональних комп'ютерах мною описано у додатку В, у таблиці В.2.

5 Виклик програмного забезпечення

Відкрити браузер, перейти у посилання: <http://international-rest.com>. Відкриється головна сторінка сайту. Користуватися вебсайтом.

6 Вхідні дані

Вхідними даними для Клієнта є:

- 1) Текстова інформація – для введення даних для реєстрації, та авторизації. Буду здійснюватися з клавіатури, або сенсорного екрану з смартфона;
- 2) Додавання елементів з меню у власне замовлення.

Вхідні дані для Менеджера:

- 1) Текстова інформація – для введення даних для реєстрації, та авторизації. Буду здійснюватися з клавіатури, або сенсорного екрану з смартфона;
- 2) Зображення у форматі .png або .jpg – додавання зображення у елемент з меню.

7 Вихідні дані

Вихідні дані для Клієнта:

- 1) Текстове відображення даних клієнта у профілі;
- 2) Інформація звіт про замовлення;

Вихідні дані для Менеджера:

- 1) Текстове зображення інформації на екрані;
- 2) Інформація Меню;
- 3) Графічне зображення елементів з Меню.