

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавр

зі спеціальності 122 «Комп'ютерні науки»,
освітньо-професійної програми «Інформаційні технології проектування»
на тему: Web-додаток підтримки роботи ресторану

Здобувача групи ІТ-92-0/2 Михайліченко Антона Сергійовича
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело.

_____ (підпис)

Антон МИХАЙЛІЧЕНКО
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник доцент кафедри ІТ, к.т.н., доц. Юлія ПАРФЕНЕНКО
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ) _____ (підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ
В.о. зав. кафедри ІТ

Світлана ВАЩЕНКО
«__» _____ 2023 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Михайліченко Антон Сергійович

1 Тема роботи Web-додаток підтримки роботи ресторану
керівник роботи Парфененко Юлія Вікторівна, к.т.н., доцент,

затверджені наказом по університету № 0588-VI від «29» травня 2023 р.

2 Строк подання студентом роботи « 07 » червня 2023 р.

3 Вхідні дані до роботи інформація для наповнення сайту, фото продуктів

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області, проектування web-додатку, розробка web-додатку

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) аналіз предметної області, актуальність, аналіз аналогів, функціональні вимоги, проектування робіт, IDEF0 - діаграми, практична реалізація, оприлюднення.

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7.Дата видачі завдання 8 лютого 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Оформлення планування робіт	24.02.2023 – 03.03.2023	
2	Оформлення технічного завдання	04.03.2023 – 08.03.2023	
3	Аналіз предметної області	09.03.2023 – 23.03.2023	
4	Проектування web-додатку	24.03.2023 – 31.03.2023	
5	Розробка web-додатку	01.04.2023 – 15.05.2023	
6	Тестування web-додатку	16.05.2023 – 23.05.2023	
7	Завантаження web-додатку на хостинг	24.05.2023 – 31.05.2023	
8	Оформлення пояснювальної записки	01.06.2023 – 07.06.2022	

Студент

(підпис)

АНТОН МИХАЙЛІЧЕНКО

Керівник роботи

(підпис)

к.т.н., доц. Юлія ПАРФЕНЕНКО

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Web-додаток підтримки роботи ресторану».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 18 найменувань, додатків. Загальний обсяг роботи – 67 сторінок, у тому числі 42 сторінки основного тексту, 2 сторінки списку використаних джерел, 23 сторінки додатків.

Кваліфікаційну роботу бакалавра присвячено розробці web-додатку підтримки роботи ресторану. Додаток, розроблений в рамках цього проекту, має потенціал для підвищення ефективності та прибутковості ресторану, а також покращення якості обслуговування клієнтів.

Перший розділ складається з огляду досліджень за тематикою роботи та аналізу аналогів розроблюваного web-додатку. Також було поставлено мету та задачі проекту.

Другий розділ включає в себе структурно-функціональне моделювання, побудову UML діаграм та бази даних. Як результат було отримано такі діаграми: контекстна діаграма IDEF0 та її декомпозиція, діаграма варіантів використання, чотири діаграми діяльності, дві діаграми послідовності.

У третьому розділі наведено та аргументовано визначену архітектуру web-додатку, описано процес розробки та використання.

В результаті проведеної роботи отримано готовий до функціонування web-додаток, який має заповнену базу даних, форму для бронювання місць, інформацію про меню та сам заклад та доменне ім'я, за допомогою якого його можна знайти в пошукових системах.

Ключові слова: web-додаток, HTML, CSS, JavaScript, ресторан, бази даних, меню, бронювання.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Огляд останніх досліджень і публікацій	8
1.2 Аналіз програмних продуктів – аналогів	9
1.3 Постановка задачі	16
1.4 Засоби реалізації web-додатку.....	17
2 ПРОЕКТУВАННЯ WEB-ДОДАТКУ.....	19
2.1 Структурно-функціональне моделювання	19
2.2 Моделювання варіантів використання web-додатку.....	23
2.3. Моделювання діаграм діяльності.....	25
2.4. Проектування бази даних web-сайту	27
3 ПРОГРАМНА РЕАЛІЗАЦІЯ WEB-ДОДАТКУ.....	30
3.1 Архітектура web-додатку.....	30
3.2. Реалізація web-додатку.....	32
3.3. Робота користувача з web-додатком.....	43
3.4. Робота адміністратора з web-додатком	46
ВИСНОВКИ	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	50
ДОДАТОК А. Технічне завдання.....	51
ДОДАТОК Б. ПЛАНУВАННЯ РОБІТ	61

ВСТУП

У сучасну епоху цифрових технологій web-додатки стали важливим інструментом для компаній, організацій і окремих людей. Оскільки все більше і більше людей використовують Інтернет для пошуку інформації, продуктів і послуг, мати web-додаток більше не розкіш, а необхідність. Web-додаток дозволяє підприємствам створювати присутність в Інтернеті, спілкуватися з клієнтами та просувати свій бренд. Web-додатки стали життєво важливою частиною сучасного життя і є важливими для всіх, хто прагне досягти успіху в сучасному цифровому світі.

Ресторани завжди відігравали важливу роль у суспільстві, пропонуючи людям місце, де вони можуть зібратися разом і насолодитися смачною їжею та компанією. Незважаючи на численні виклики, з якими зіткнулася ресторанна індустрія в останні роки, ресторани залишаються актуальними і в наш час, а їх важливість тільки зростає. Ресторани дають можливість спробувати нові кухні та смаки, скуштувати різні страви та насолодитися кулінарними шедеврами, які нелегко отримати деінде. Ресторан – це бізнес, який у сучасну цифрову епоху повинен мати сильну присутність в Інтернеті. Наявність web-додатку є одним з найважливіших кроків, які ресторан може зробити для встановлення своєї присутності в Інтернеті та охоплення ширшої аудиторії.

Web-додаток надає можливість продемонструвати меню ресторану, його місцезнаходження, години роботи та контактну інформацію. Маючи власний web-додаток, ресторан може створити унікальну ідентичність бренду, яка відрізняє його від конкурентів і приваблює клієнтів.

Однією з найбільших переваг наявності web-додатку для ресторану є можливість здійснювати онлайн-замовлення. Надаючи онлайн меню і систему замовлень, ресторан може полегшити клієнтам замовлення їжі з будь-якого

місця і в будь-який час. Це не тільки підвищує фактор зручності для клієнтів, але й дозволяє ресторанам оптимізувати свої операції та більш ефективно управляти замовленнями.

Web-додаток також є платформою для взаємодії ресторану зі своїми клієнтами. Використовуючи такі функції, як відгуки клієнтів, інтеграція з соціальними мережами та онлайн-бронювання, ресторан може створити лояльну аудиторію та покращити свою репутацію. Це допомагає створити кращий досвід для клієнтів і підвищує ймовірність повторних відвідувань.

Мета кваліфікаційної роботи бакалавра – створення web-додатку підтримки роботи ресторану.

Для досягнення мети потрібно виконати такі задачі:

- проведення дослідження впровадження інформаційних технологій для підтримки роботи ресторану;
- аналіз продуктів аналогів та формулювання функціональних вимог, розробка плану та графіка робіт, проектування UI/UX;
- проектування web-додатку;
- розробка web-додатку та його тестування.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Існує багато досліджень і публікацій на тему підтримки web-додатків для роботи ресторанів, в яких основна увага приділяється різним технологіям і підходам, що використовуються при створенні web-додатків для управління ресторанами, а також ефективності та впливу цих додатків на роботу ресторану і задоволеність клієнтів.

В одному з останніх досліджень, опублікованому в Міжнародному журналі комп'ютерних наук та інформаційних технологій у 2021 році, було проаналізовано ефективність web-додатку для управління рестораном, який включав такі функції, як управління замовленнями, управління запасами та управління клієнтами. Дослідження показало, що використання web-додатку підвищило ефективність і точність роботи ресторану, а також підвищило рівень задоволеності клієнтів [1].

Інше дослідження, опубліковане в роботі [2], вивчало використання мобільних додатків для управління ресторанами і показало, що мобільні додатки забезпечують більшу гнучкість і зручність для операторів ресторанів, а також підвищують задоволеність клієнтів завдяки більш персоналізованому та ефективному обслуговуванню.

На додаток до академічних досліджень, існують публікації та онлайн-ресурси, доступні для web-розробників та операторів ресторанів, які прагнуть створити та впровадити web-додатки для управління ресторанами. Одним з таких ресурсів є Restaurant Engine – web-додаток, який надає інформацію та інструменти для створення web-додатків ресторанів та систем онлайн-замовлень [3].

Загалом, дослідження та публікації про web-додатки для управління ресторанами свідчать про те, що ці додатки можуть підвищити ефективність та задоволеність клієнтів і стають все більш важливим інструментом для операторів ресторанів у сучасному цифровому ландшафті.

1.2 Аналіз програмних продуктів – аналогів

Усі найкращі web-додатки ресторанів мають одну спільну рису: вони спрощують онлайн-відвідувачам пошук інформації та оформлення замовлення. Зрозуміло, що добре розроблений web-додаток ресторану є ключовим для того, щоб залучити клієнтів (або замовити онлайн), але створити web-додаток ресторану не дуже легка справа [5]. Для порівняння було обрано декілька закордонних ресторанів, що мають дуже високий рейтинг в світі та слугують прикладом для багатьох рестораторів.

Першим додатком для порівняння було обрано The Original, закусочна в Денвері від Sage Restaurant Concepts [6]. Ця місцева їдальня, побудована на ідеї переосмислити місце, куди ходять місцеві жителі, задовольнить відвідувачів творчими акцентами. Їхній web-додаток є прикладом того, як перенести ресторанний досвід в Інтернет, із професійними ілюстраціями страв, які оживляють web-додаток. Темно-синя палітра кольорів (рис. 1.1) у web-додатку дозволяє гостям зосередитися під час перегляду меню (рис. 1.2) та нових подій у ресторані, сторінка з геоданими надає детальну інформацію про місцезнаходження (рис. 1.3). Виділена кнопка «Бронювання» оптимізована, щоб спонукати відвідувачів забронювати столик, таким чином перетворюючи онлайн гостя на клієнта.

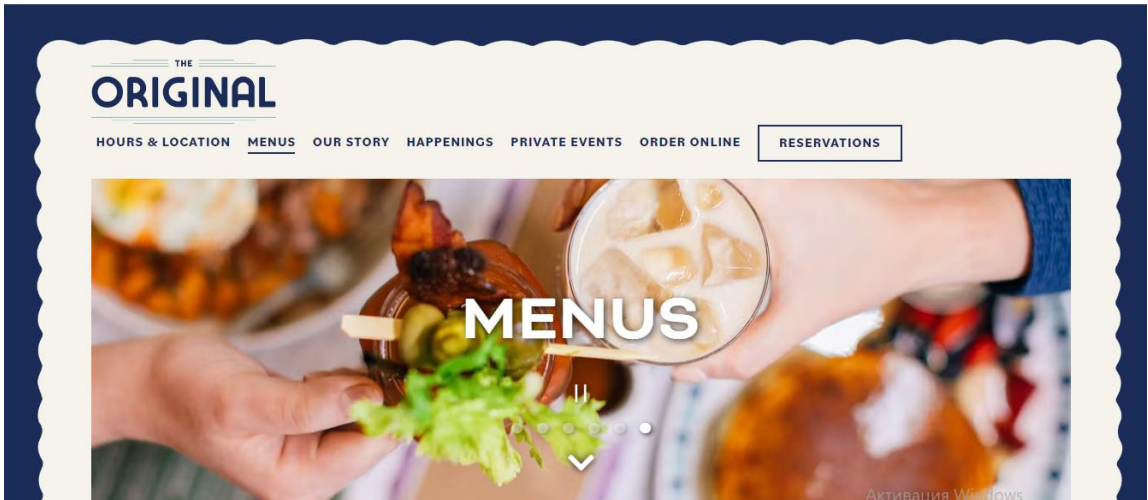


Рисунок 1.1 – Головна сторінка

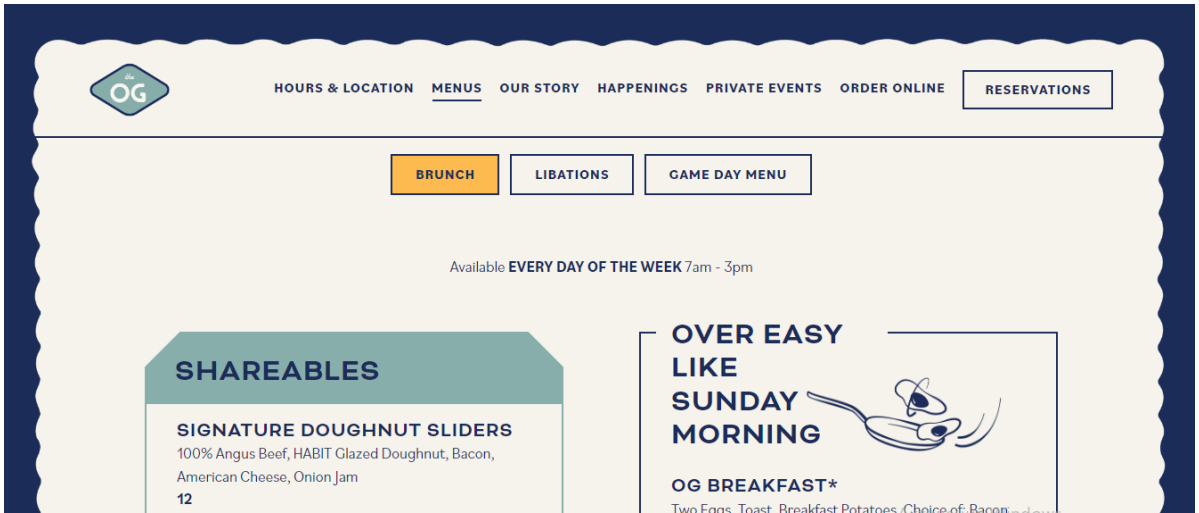


Рисунок 1.2 – Сторінка меню

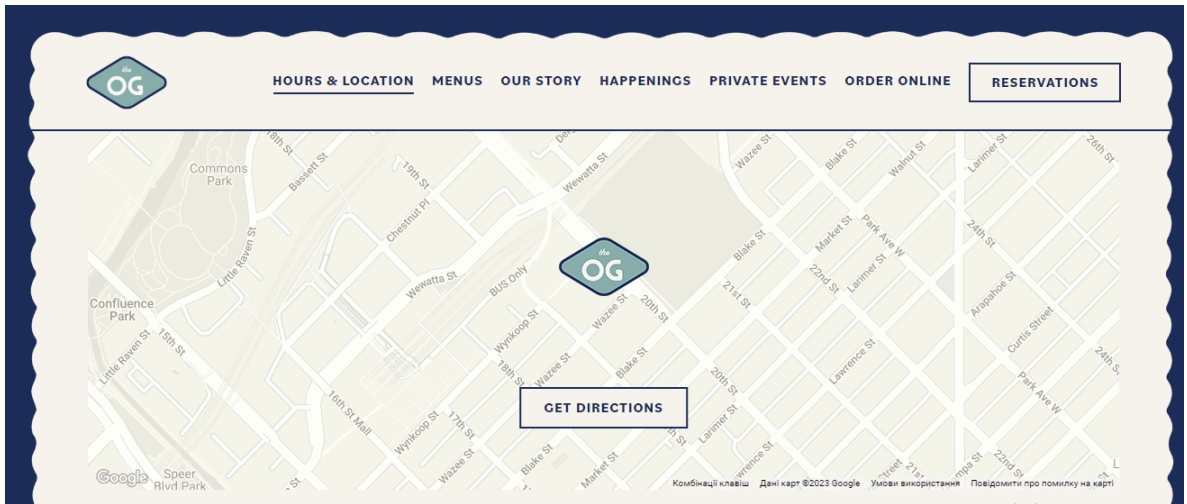


Рисунок 1.3 – Геололокація на web-додатку

Наступним рестораном було обрано Federalist Pig [7], що є одним з найвідоміших закладів барбекю у Вашингтоні, округ Колумбія. Головна сторінка його web-додатку (рис. 1.4) є платформою, спрямованою на емоційне враження користувачів за допомогою використання насичених кольорів, провокаційних ліній ілюстрацій та привабливих зображень страв (рис. 1.5). Незалежно від високої естетичної привабливості системи дизайну, сам web-додаток відрізняється грамотним структуруванням і зручною системою навігації для користувачів (рис. 1.6).

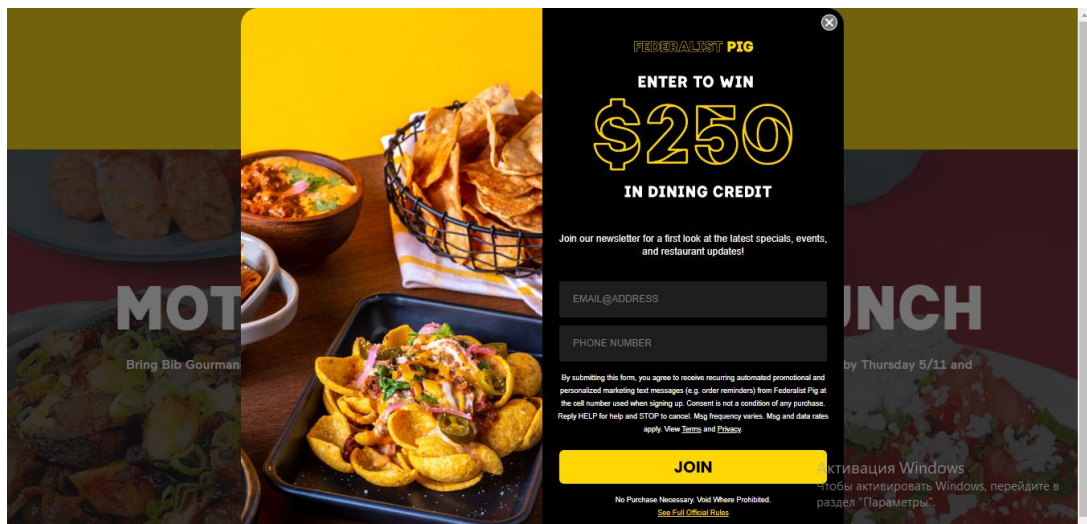


Рисунок 1.4 – Головна сторінка на web-додатку

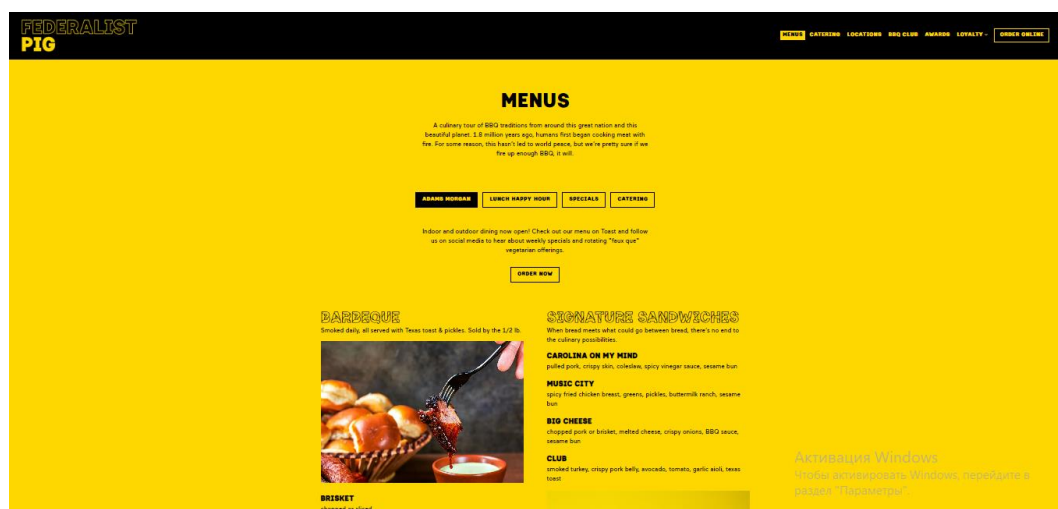


Рисунок 1.5 – Реалізація меню на web-додатку

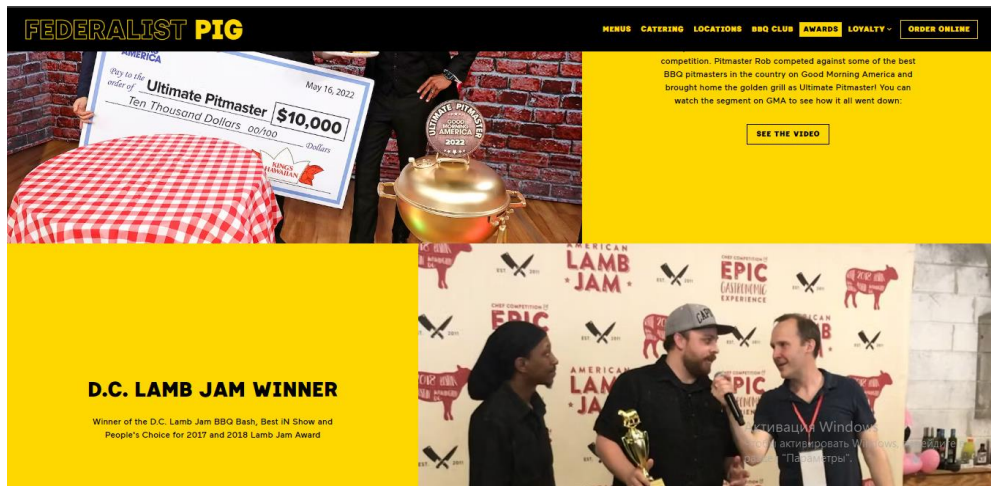


Рисунок 1.6 – Навігація на web-додатку

Третім web-додатком було обрано Anton's [8] – це ностальгічне нью-йоркське кафе, кухня якого відображає історію Нью-Йорка з глибокими європейськими впливами. Їхній web-додаток викликає відчуття неокласицизму завдяки використанню курсивних шрифтів із зарубками, а також красивих фотографій кухні та інтер'єрів. Використання Anton's бічної навігації (рис. 1.7) дозволяє відвідувачам швидко переглядати меню (рис. 1.8), години роботи та статті в пресі. На сторінках web-додатку використовуються виділені кнопки як заклик до дії, тому гості можуть швидко забронювати столик (рис. 1.9) або навіть підписатися на електронну розсилку.

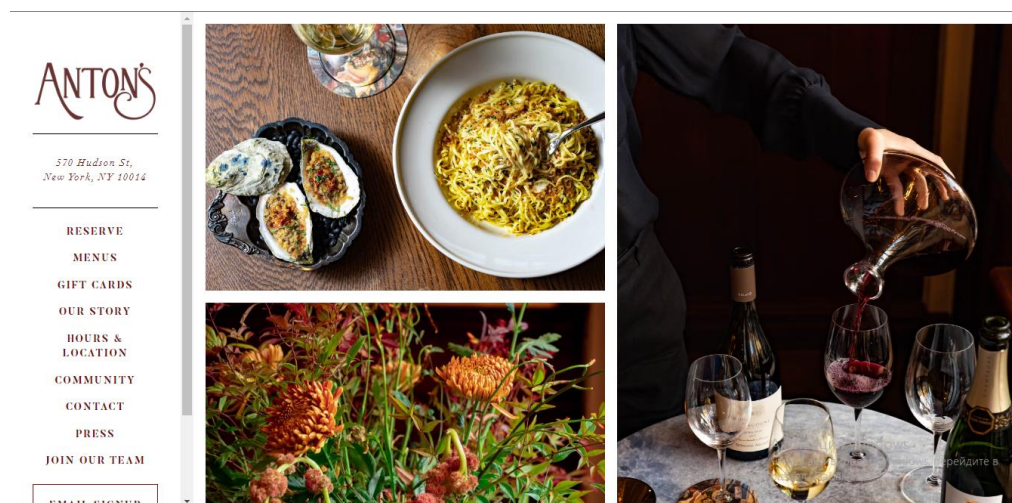


Рисунок 1.7 – Головна сторінка на web-додатку

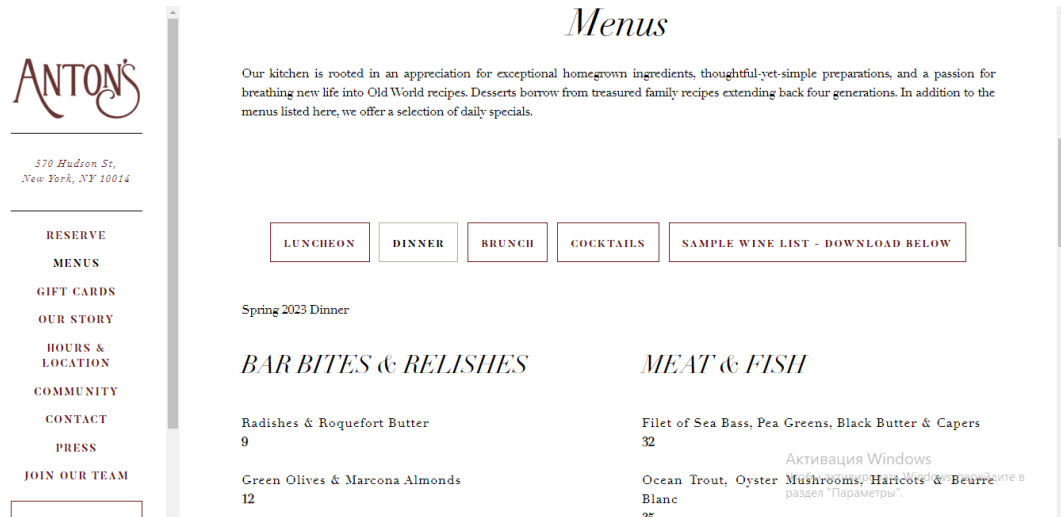


Рисунок 1.8 – Вкладка меню на web-додатку

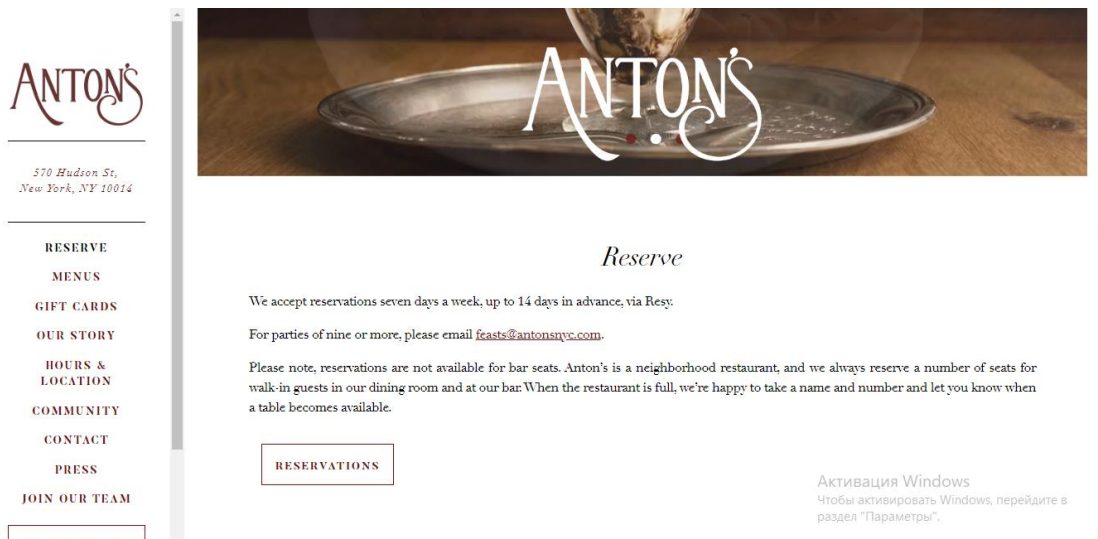


Рисунок 1.9 – Сторінка резервування

Останнім web-додатком було обрано Damian [9], що є концепцією-близнюком знаменитого нью-йоркського Cosme і має міцне коріння в мексиканській культурі та кухні, з акцентом на сезонності та використанні найкращих каліфорнійських продуктів. Шеф-кухар-власник ресторану Енріке Ольвера застосовує новий підхід до того, як гості відчують мексиканську кухню в ресторані Damian, що відображається у web-додатку. Навігація додатком відбувається простим скролінгом (рис. 1.10), а розділ "Про нас", меню страв і напоїв (рис. 1.11-1.12) та критично важлива інформація про

місцезнаходження знаходяться в одному просторі (рис. 1.13). Незалежно від того, чи потрібно зарезервувати столик або придбати цифрову подарункову картку, відвідувачі можуть зробити це на web-додатку з легкістю.



Рисунок 1.10 – Сторінка завантаження web-додатку



Рисунок 1.11 – Візуалізація меню



Рисунок 1.12 – Вкладка меню на web-додатку

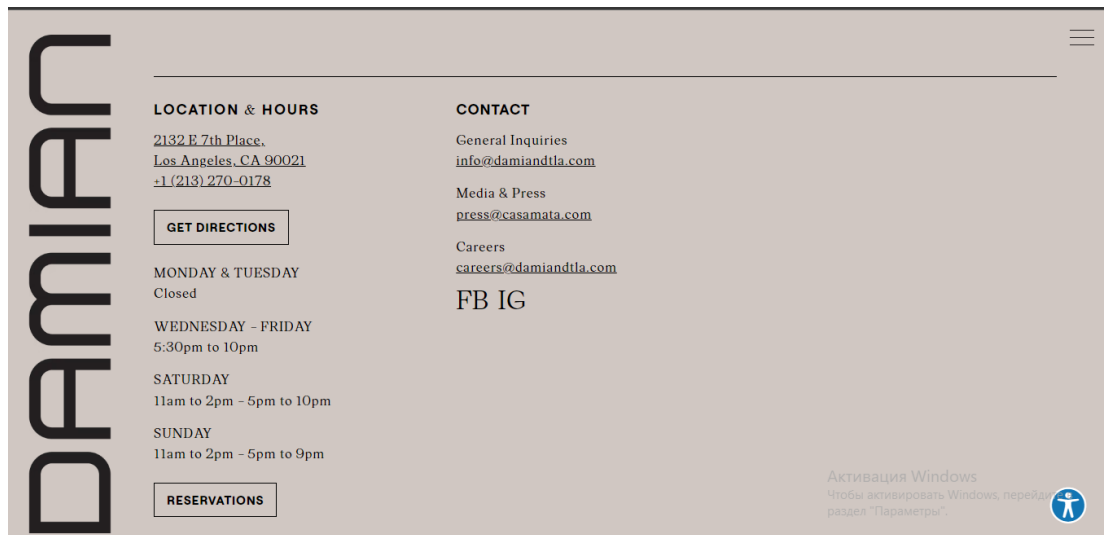


Рисунок 1.13 – Інформація про заклад

Таблиця 1.1 – Порівняння web-додатків ресторанів

Критерії	The Original	Federalist Pig	Anton's	Damian
Зручність використання web-додатку	+	+	+	+
Дизайн і візуальна привабливість	+	+	+	+

Змістовність	+	-	-	+
Відгуки про заклад	+	-	-	-
Навігація	+	+	+	+
Адаптивність	+	+	+	+
Швидкість та продуктивність	-	-	-	-
SEO-оптимізація	+	-	-	+
Наявність контактів для зв'язку	+	+	+	+

Всі розглянуті заклади мають доволі зручний функціонал та графічне оформлення. З недоліків можна зазначити, що не всі web-додатки мають візуалізацію страв, що може змусити клієнтів обрати більш передбачуваний заклад, також резервування є не доволі зрозумілим, треба докласти зусиль, щоб все зробити без помилок.

1.3 Постановка задачі

Ресторанна індустрія – це висококонкурентний і динамічний сектор, в якому швидко змінюються споживчі вподобання та зростають очікування щодо швидкого і зручного обслуговування. Однак багато ресторанів все ще покладаються на застарілі ручні процеси та системи, що може призвести до неефективності, помилок та незадоволення клієнтів.

Однією з ключових проблем, з якою сьогодні стикаються керівники ресторанів, є необхідність керувати широким спектром завдань одночасно - від прийому замовлень і обробки платежів до управління запасами і складання

розкладу роботи персоналу. Це вимагає високого рівня координації та комунікації між різними відділами та співробітниками, чого може бути важко досягти, використовуючи традиційні паперові системи або ручні процеси.

Таким чином, проблема, яку намагається вирішити ця кваліфікаційна робота, полягає в необхідності створення web-додатку, який може забезпечити комплексну платформу для управління всіма аспектами діяльності ресторану, включаючи онлайн-замовлення та бронювання місць. Web-додаток буде спрямований на підвищення ефективності, продуктивності та прибутковості роботи ресторану шляхом надання централізованої платформи для управління всіма завданнями та процесами, покращення комунікації та координації між співробітниками.

Метою проєкту є створення web-додатку підтримки роботи ресторану, тобто спрощення умови бронювання столиків та надання інформації про мережу ресторанів.

Web-додаток буде надавати спрощення умови бронювання столиків та надавати інформацію про мережу ресторанів.

База даних буде містити інформацію про бронювання, страви з відповідними фотокартками та категорії меню.

Технічне завдання по розробці web-додатку наведене в Додатку А, планування робіт в Додатку В.

1.4 Засоби реалізації web-додатку

Існують різні способи реалізації web-додатків для підтримки роботи ресторанів. Даний web-додаток передбачає розробку мовою Python. Одним із варіантів є використання фреймворку для веб-розробки, такого як Django, для створення додатку [10]. Цей фреймворк надає набір інструментів і бібліотек,

які спрощують процес розробки і дозволяє створювати масштабовані і підтримувані web-додатки.

При розробці web-додатку для підтримки роботи ресторану використовується база даних SQLite для управління даними, пов'язаними з пунктами меню закладу, бронюванням клієнтів та інформацією про сам заклад.

Задання стилів буде здійснюватися за допомогою CSS, а розмітка сторінки – HTML.

Мовою програмування JavaScript було додано анімованість елементам.

2 ПРОЕКТУВАННЯ WEB-ДОДАТКУ

2.1 Структурно-функціональне моделювання

Методологія IDEF0 є широко використовуваною технікою моделювання в галузі системної та програмної інженерії [11]. Вона забезпечує структурований підхід для фіксації, аналізу та документування функцій і процесів системи.

Основною метою IDEF0 є створення чітких і стислих моделей, які представляють функції, входи, виходи, елементи управління і механізми системи або процесу. Він використовує ієрархічну структуру для розбиття складних систем на більш керовані компоненти, що дозволяє краще зрозуміти і проаналізувати їх.

Ключові компоненти IDEF0:

- Функція: В IDEF0 функція представляє дію або процес, який перетворює вхідні дані у вихідні. Це фундаментальний будівельний блок моделі і представлений на діаграмі у вигляді рамки. Функції описуються за допомогою дієслів дії і забезпечують чітке розуміння завдань, що виконуються системою.

- Вхід і вихід: Кожна функція має входи і виходи, які визначають інформацію або матеріали, необхідні для виконання функції, і результати, які вона виробляє, відповідно. Входи і виходи зображені стрілками, що з'єднують функції, і позначені, щоб вказати характер даних або матеріалів, що передаються.

- Контроль: Механізми управління визначають умови або правила, за якими виконується функція. Вони визначають, коли функція починається, зупиняється або продовжується на основі певних критеріїв. Контроль зображується пунктирною лінією, що з'єднує механізм контролю з функцією.

- Механізм: Механізми представляють ресурси, інструменти або обладнання, необхідні для виконання функції. Вони включають фізичні або логічні сутності, які сприяють виконанню функції. Механізми зображуються у вигляді кіл, з'єднаних з функцією.

IDEF0 забезпечує стандартизовану та візуальну мову моделювання, яка полегшує чітку комунікацію, аналіз, інтеграцію та прийняття рішень для складних систем та процесів.

Контекстну діаграму процесу підтримки роботи ресторану показано на рис. 2.1. На вхід IDEF0 поступає запит користувача. Серед механізмів було зазначено користувача та базу даних. Керується функція документацією. На виході отримуємо підтверджене бронювання.

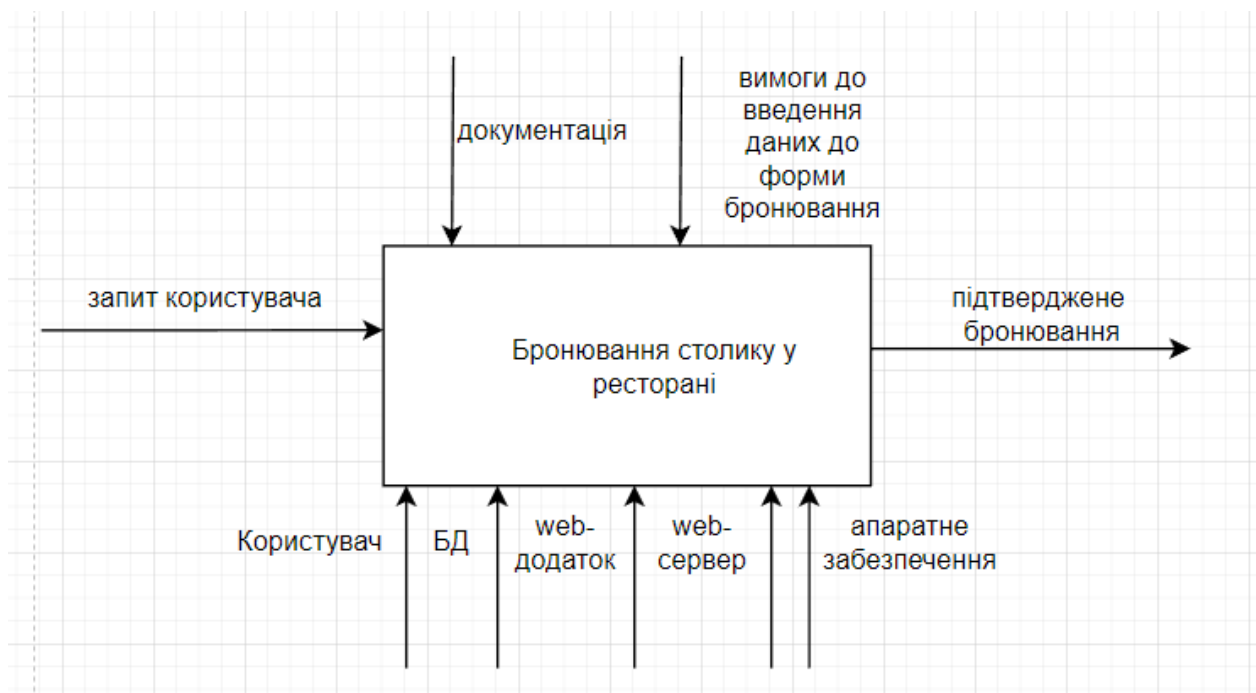


Рисунок 2.1 – Контекстна діаграма процесу бронювання столику за допомогою web-додатку

Таблиця 2.1 – Дані для діаграми декомпозиції

Підпроцес	Вхід	Управління	Механізм	Вихід
Перехід на сайт	Запит користувача	документація	Користувач, БД	перехід між сторінками
Навігація по сайту	перехід між сторінками	документація, форма бронювання	Користувач, БД	перехід на форму бронювання
Заповнення інформації для бронювання	перехід на форму бронювання	документація, форма бронювання	Користувач, БД	надсилання форми
Оформлення бронювання	надсилання форми	документація	Користувач, БД	підтвержене бронювання

На рисунку 2.2 показано діаграму декомпозиції першого рівня.

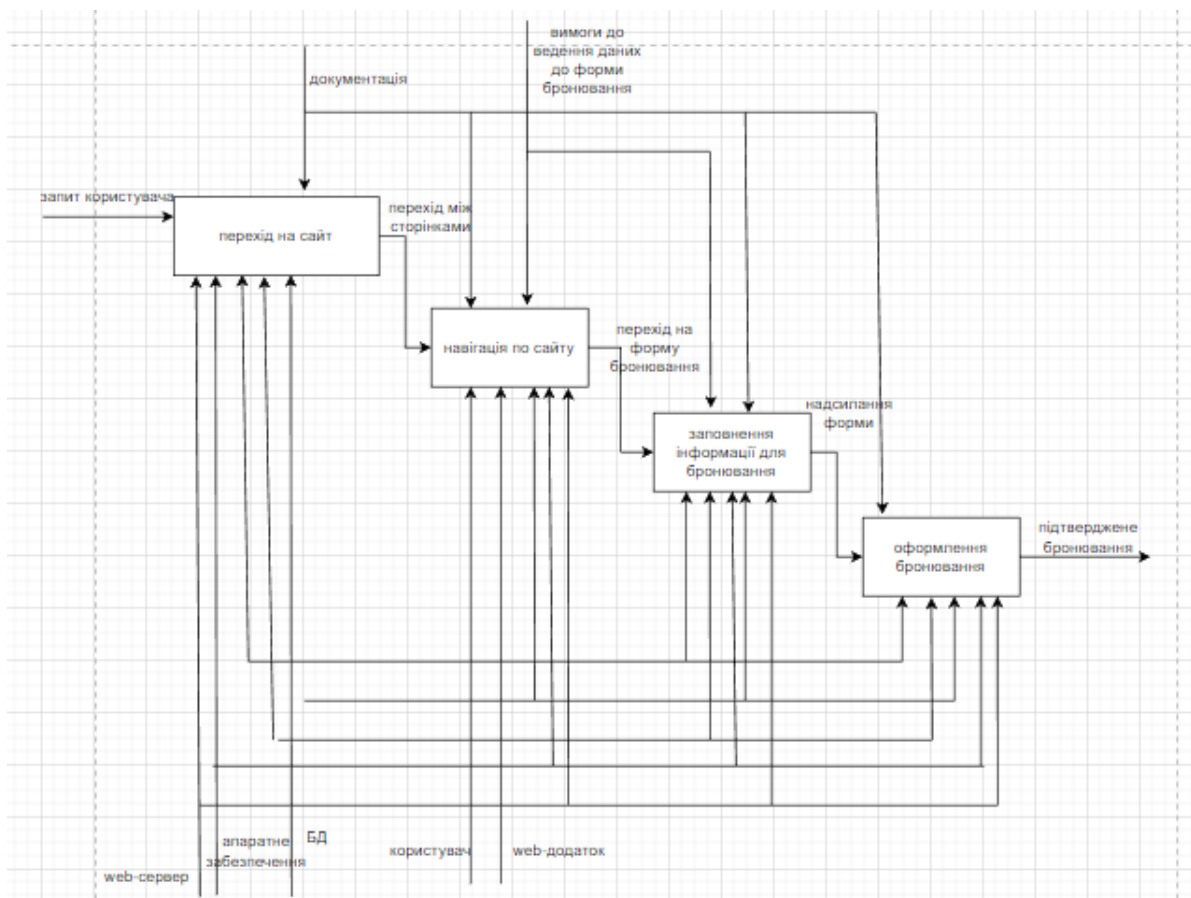


Рисунок 2.2 – Діаграма декомпозиції першого рівня

Блок “ перехід на сайт”:

- Вхідною стрілкою є “запит користувача”;
- Вихідною стрілкою – “перехід між сторінками”;
- Стрілка контролю – “документація”;
- Стрілки механізму – “Користувач”, “ web-додаток ”, “ БД ”, “web-сервер”, “Апаратне забезпечення”.

Блок “ навігація по сайту”:

- Вхідною стрілкою є “ перехід між сторінками”;
- Вихідною стрілкою – “ перехід на форму бронювання”;
- Стрілка контролю – “документація”, “вимоги до ведення даних до форми бронювання”;
- Стрілки механізму – “Користувач”, “ web-додаток ”, “ БД ”, “web-сервер”, “Апаратне забезпечення”.

Блок “ заповнення інформації для бронювання”:

- Вхідною стрілкою є “ перехід на форму бронювання”;
- Вихідною стрілкою – “ надсилання форми”;
- Стрілка контролю – “документація”, “вимоги до ведення даних до форми бронювання”;
- Стрілки механізму – “Користувач”, “ web-додаток ”, “ БД ”, “web-сервер”, “Апаратне забезпечення”.

Блок “ оформлення бронювання”:

- Вхідною стрілкою є “ надсилання форми”;
- Вихідною стрілкою – “ підтверджене бронювання”;
- Стрілка контролю – “документація”;
- Стрілки механізму – “Користувач”, “ web-додаток ”, “ БД ”, “web-сервер”, “Апаратне забезпечення”.

2.2 Моделювання варіантів використання web-додатку

Діаграма варіантів використання – це візуальне представлення, яке відображає взаємодію між акторами (користувачами або зовнішніми системами) і самою системою [19]. У контексті web-додатків, призначених для підтримки роботи ресторану, діаграма варіантів використання допомагає визначити різні функціональні можливості і те, як різні користувачі або системи взаємодіють з додатком. У цьому розділі представлені ключові актори (табл. 2.2) та відповідні їм варіанти використання (табл. 2.3) на діаграмі варіантів використання для web-додатку ресторану.

Таблиця 2.2 – Ключові актори

№	Актор	Опис
1.	Клієнт	Клієнт є основним актором, який взаємодіє з web-додатком для бронювання, надання зворотного зв'язку та доступу до інших функцій, пов'язаних з клієнтом.
2.	Адміністратор	Цей актор представляє працівників ресторану, які використовують web-додаток для виконання таких завдань, як управління меню, резервування місць та оновлення інформації про ресторан. Також має доступ до адміністративних функцій, які контролюють роботу веб-додатку.
3.	База даних	Актор, що зберігає інформацію про бронювання та інформацію про категорію меню та окремі страви.

Таблиця 2.3 – Опис варіантів використання

№	Назва	Опис
1.	Авторизація	Надає доступ адміністраторам до функціоналу
2.	Бронювання	Клієнт може зробити бронювання через web-додаток, вказавши такі дані, як дата, час, кількість гостей та будь-які особливі вимоги.
3.	Управління меню	Персонал ресторану може додавати, оновлювати або видаляти позиції з меню через web -додаток. Вони можуть вказати такі деталі, як назва позиції, опис, ціна та наявність.
4.	Надання інформації	Адміністратор надає актуальну інформацію про заклад.
5.	Перегляд сторінок	Користувачі мають змогу переглядати інформацію, що їх цікавить.

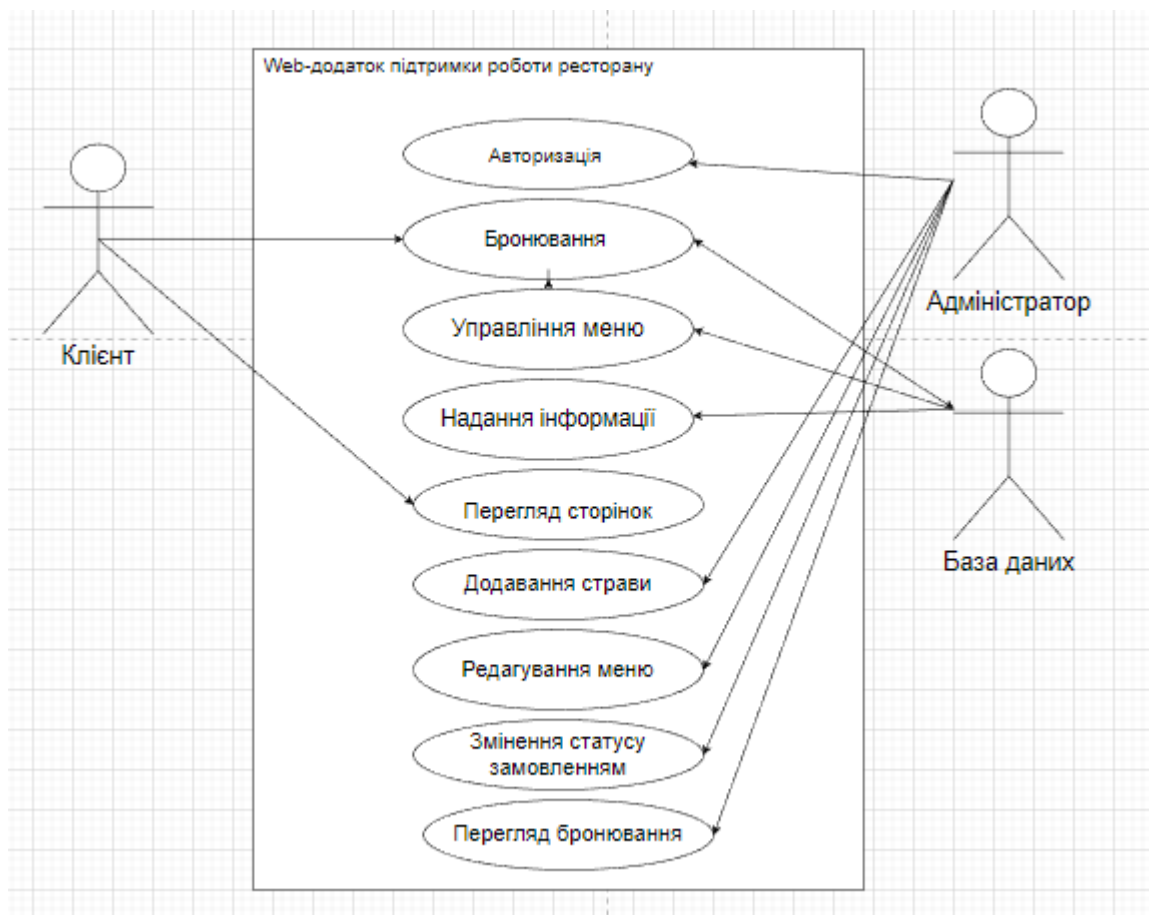


Рисунок 2.3 – Діаграма варіантів використання web-додатку

Діаграма варіантів використання надає всебічний огляд функціональних можливостей і взаємодій у веб-додатку для ресторану. Актор «Клієнт» має змогу оформлювати бронювання та переглядати сторінки. За допомогою актора «Бази даних» є можливість взаємодії з бронюванням, редагуванням меню та зберіганням інформації. Актор «Адміністратор» має змогу авторизації, редагування даних та зміни статусу бронювання. Вона допомагає зрозуміти системні вимоги, визначити ролі користувачів і розробити функції додатку для забезпечення оптимального користувацького досвіду та ефективної роботи ресторану.

2.3. Моделювання діаграм діяльності

Діаграми діяльності – це інструмент моделювання, який використовується для візуального представлення потоку діяльності та послідовності дій у системі або процесі [20]. У контексті web-додатку, розробленого для підтримки мережі ресторанів, діаграми діяльності є цінними для розуміння та ілюстрації різних видів діяльності та їх взаємозв'язків.

На рисунку 2.4 зображено діаграму діяльності модулю вибору страви.

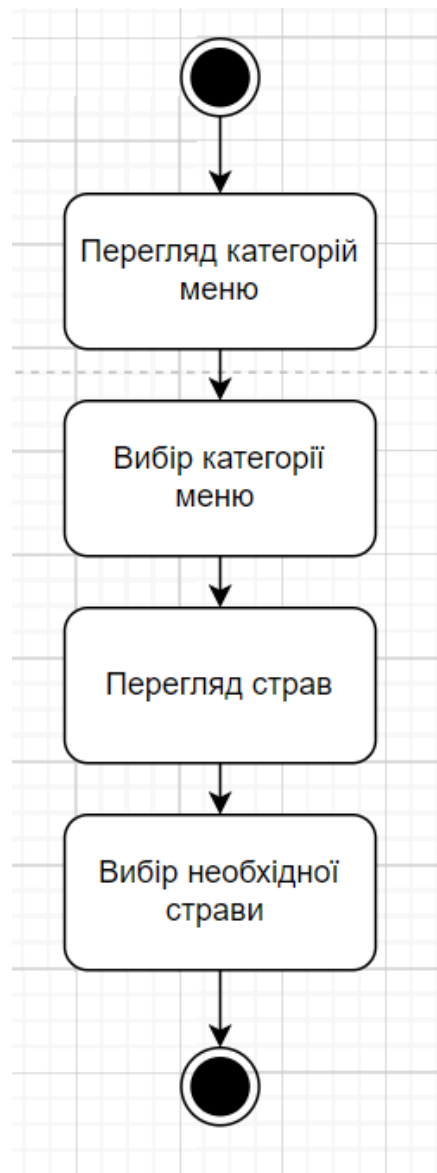


Рисунок 2.4 – Діаграма діяльності модулю вибору страви

На рисунку 2.5 зображено діаграму діяльності модулю редагування бронювання.

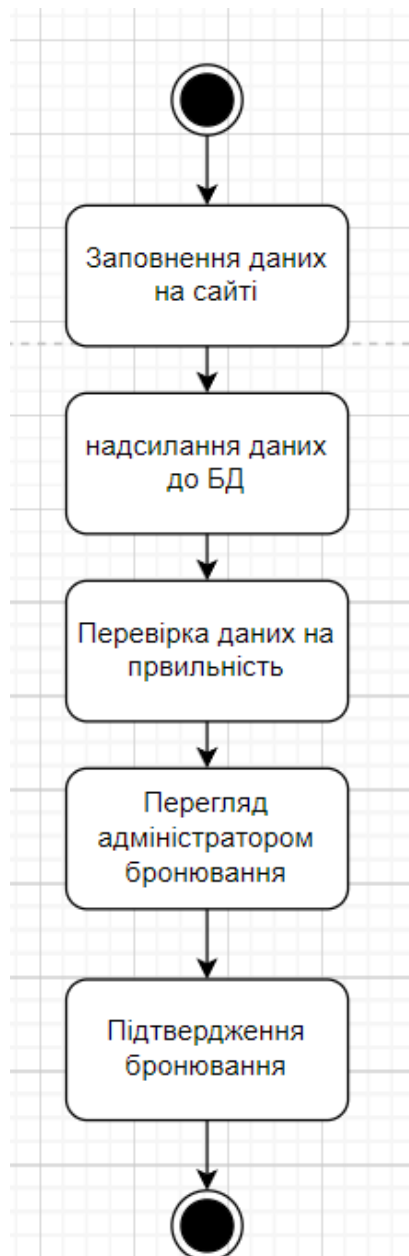


Рисунок 2.5 – Діаграма діяльності модулю редагування бронювання

2.4. Проектування бази даних web-сайту

Web-додаток має базу даних, що складається з трьох таблиць, що містять в собі інформацію (рис. 2.6-2.7), таку як бронювання, аутентифікацію та інформацію про меню та його категорії.

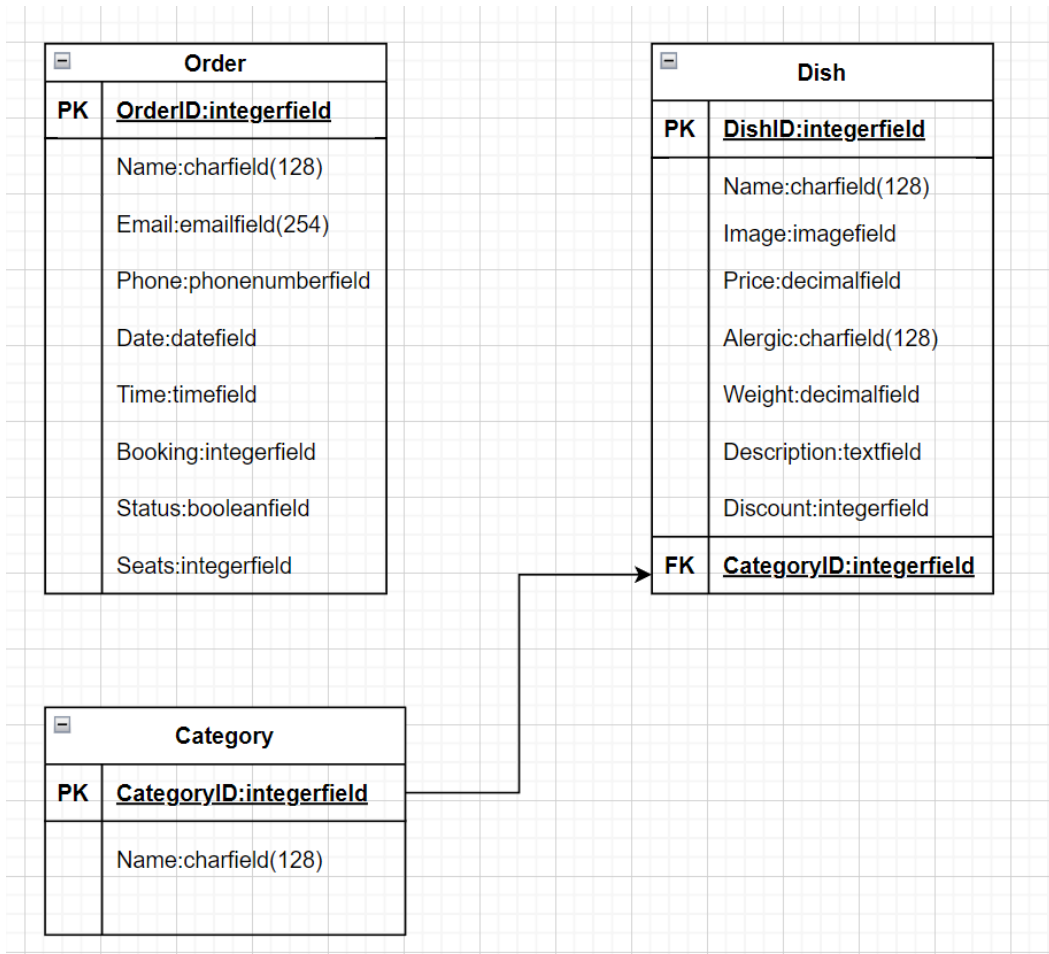


Рисунок 2.6 – Схема бази даних

Таблиця 2.4 – Опис значень полів бази даних

№	Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
1	Order	<u>OrderID</u>	Ідентифікатор бронювання	<u>integerfield</u>	PK	
		Name	ПІБ замовника	charfield(128)		
		Email	Електронна пошта	emailfield(254)		
		Phone	Номер телефону	phonenumbersfield		
		Date	Дата бронювання	datefield		
		Time	Час	timefield		
		Booking	Кількість годин бронювання	integerfield		
		Status	Статус замовлення	booleanfield		
		Seats	Кількість місць	integerfield		
2	Category	<u>Category ID</u>	Ідентифікатор категорії	<u>integerfield</u>	PK	
		Name	Назва категорії	charfield(128)		
3	Dish	<u>DishID</u>	Ідентифікатор страви	<u>integerfield</u>	PK	
		Name	Назва страви	charfield(128)		
		Image	Фотографія страви	imagefield		
		Price	Ціна страви	decimalfield		
		Alergic	Вміст алергену	charfield(128)		
		Weight	Вага страви	decimalfield		
		Description	Опис страви	textfield		
		Discount	Знижка на страву	integerfield		
		<u>CategoryID</u>	Категорія до якої належить страву	<u>integerfield</u>	FK	

3 ПРОГРАМНА РЕАЛІЗАЦІЯ WEB-ДОДАТКУ

3.1 Архітектура web-додатку

Модель MVC - це архітектурний шаблон програмного забезпечення, який розділяє додаток на три основні компоненти. В даному випадку модель представляє дані та бізнес-логіку, подання відображає користувацький інтерфейс, а контролер обробляє вхідні дані користувача, оновлює модель та керує зв'язком між моделлю та поданням.

Архітектура додатку (рис. 3.1) була обрана клієнт-серверна, вона складається з реляційної бази даних та серверу бази даних Django.

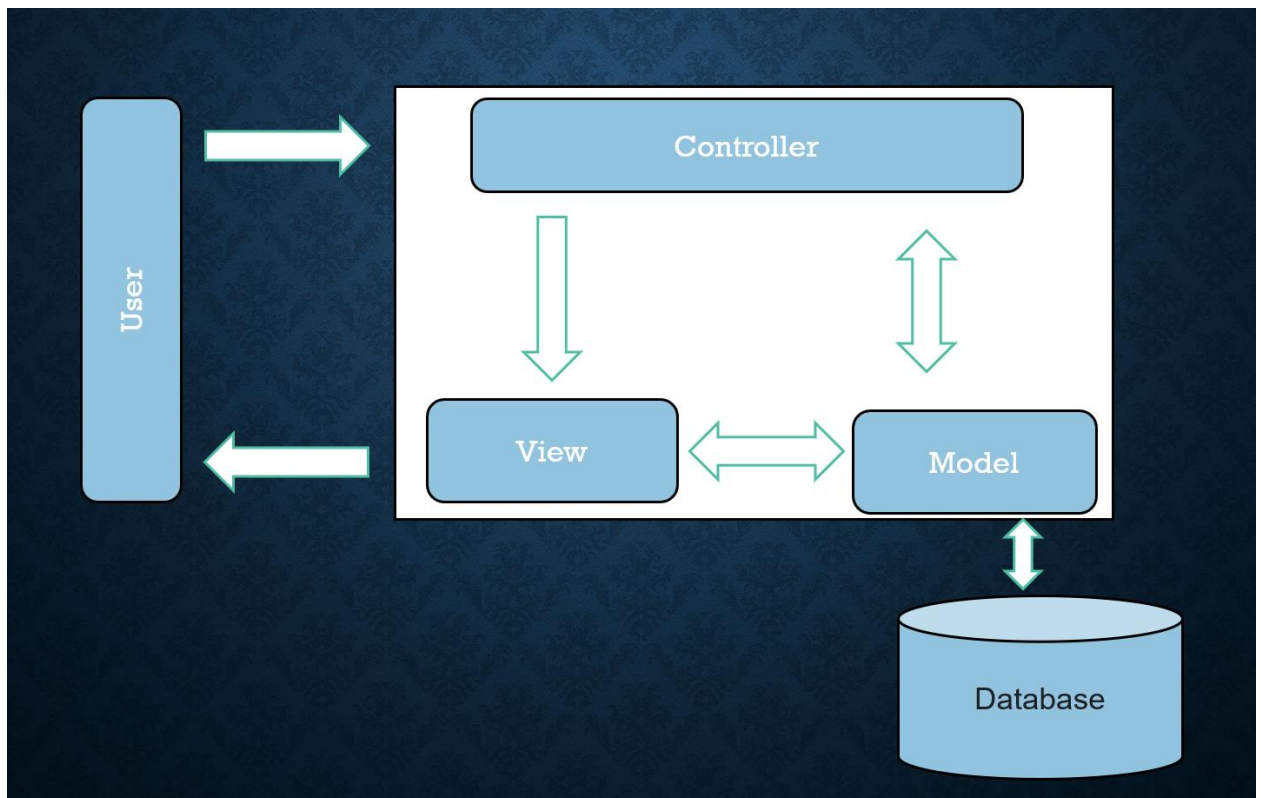


Рисунок 3.1 – Архітектура web-додатку

Фронтенд додатку відповідає за користувацький інтерфейс та взаємодію з користувачем. Він побудований з використанням сучасних веб-технологій,

таких як HTML, CSS та JavaScript. Фронтенд складається з декількох сторінок і компонентів для реалізації різних функцій.

Головна сторінка: ця сторінка містить загальну інформацію про ресторан, таку як його назва, кнопку, що надає можливість замовити столик та фотографії закладу.

Форма бронювання: Ця форма дозволяє відвідувачам сайту заповнити її для бронювання місця в ресторані. Вона збирає таку інформацію, як ім'я відвідувача, контактні дані, бажані дата і час, а також кількість бажаних місць.

Інформаційна сторінка: На цій сторінці відображається детальна інформація про ресторан, включаючи фото персоналу та інформацію про них.

Сторінка Меню передбачає візуалізацію всіх доступних страв, що є актуальними в даний період. В кожній страві буде міститися його фотографія, опис, наявність алергенів та назва.

Сторінка з контактною інформацією показує відвідувачам місцезнаходження на карті, номери телефонів та пошта, яка актуальною для звернень.

Внутрішня частина додатку відповідає за логіку на стороні сервера, операції з базою даних та автентифікацію.

Веб-сервер: Веб-сервер Nginx, обробляє вхідні HTTP-запити від клієнтів і перенаправляє їх до відповідних компонентів програми.

Автентифікація: Додаток має механізм автентифікації, щоб гарантувати, що тільки адміністратор має доступ до адміністративних функцій. Це досягнуто за допомогою таких методів, як автентифікація за допомогою імені користувача та пароля.

База даних: Додаток має базу даних для зберігання інформації про бронювання, деталі меню. Для цього використано реляційну систему управління базами даних (СУБД), таку як SQLite

База даних зберігає відповідну інформацію, необхідну додатку. Вона має кілька таблиць для зберігання таких даних, як бронювання, аутентифікацію та інформацію про меню та його категорії.

Таблиця бронювань: Ця таблиця зберігає дані про бронювання, зроблені відвідувачами сайту. Вона містить такі поля, як ідентифікатор бронювання, ім'я відвідувача, номер телефону, пошта, бажані дата і час, а також кількість місць.

Загалом, така архітектура дозволяє адміністратору керувати інформацією про ресторан, відвідувачам бронювати місця через форму, а працівникам підтверджувати бронювання. Відвідувачі також можуть переглядати інформацію про заклад через додаток.

3.2. Реалізація web-додатку

За основну мову було взято Python, а фреймворк внутрішнього програмування Django [10]. Було встановлено систему управління базами даних SQLite (рис. 3.2).

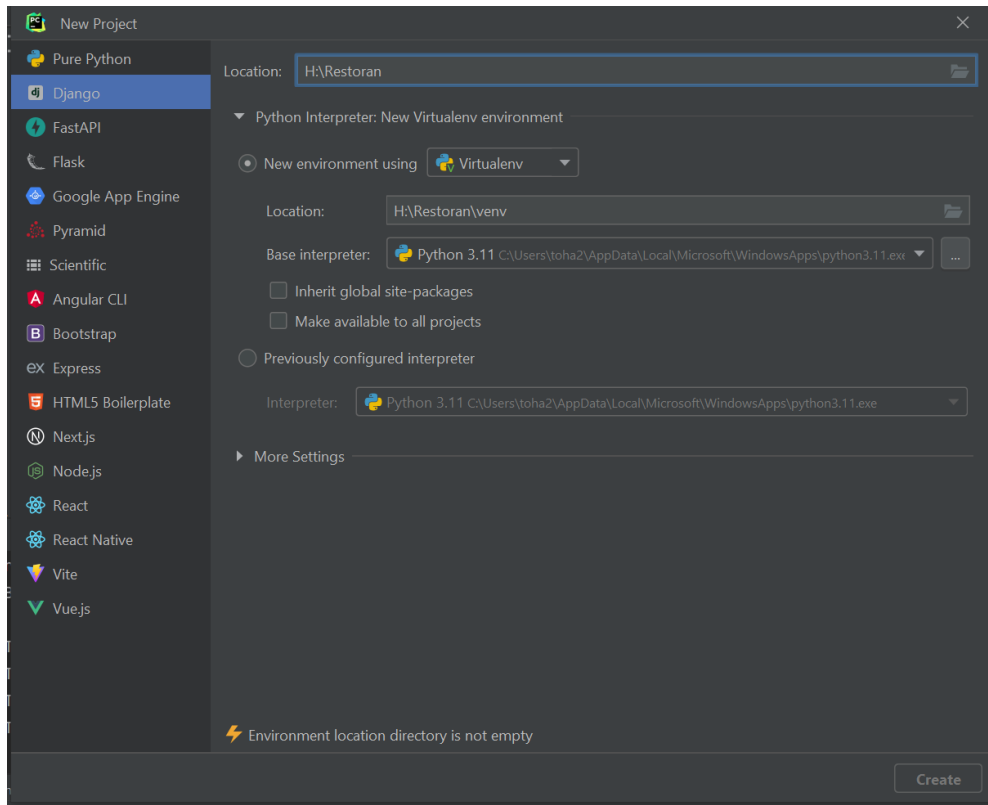


Рисунок 3.2 – Створення проекту на Django

Першочергово було налаштовано середовище розробки: у файлі `settings.py` та створено папки файлів сторінок (`About`, `Change`, `Contact`). Файли `Views` в Django використовуються для обробки запитів, обробки бізнес-логіки та відправлення відповідей. Файл `Urls` в Django використовується для визначення маршрутів, які вказують, які `views` будуть викликатися для обробки конкретних запитів. Основна функція файлу `Models` в Django полягає в тому, щоб визначити класи моделей, які відображають таблиці в базі даних (рис. 3.3).

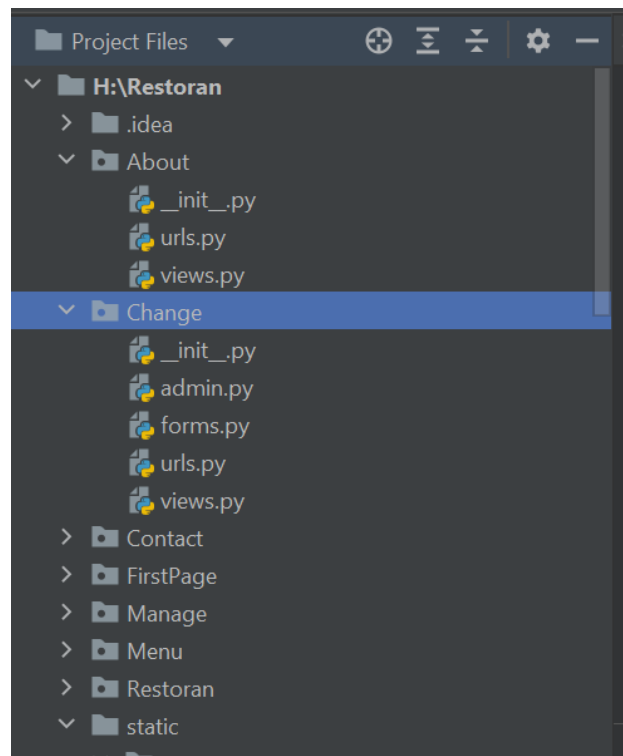


Рисунок 3.3 – Папки файлів сторінок

Потім було налаштовано середовище розробки для фронтенд-технологій, таких як CSS.

Проектування бази даних з таблицями Orders, Dishes та Categories. Таблиця Orders містить такі елементи: ім'я, електронна пошта, номер, дата та час та кількість місць (рис. 3.4).

The image shows a dark-themed web form titled "Add Order". It contains the following fields and controls:

- Name:** A text input field.
- Email:** A text input field.
- Phone:** A text input field.
- Date:** A date picker with a "Today" button and a calendar icon.
- Time:** A time picker with a "Now" button and a clock icon.
- Note:** A small text note below the date and time pickers: "Note: You are 3 hours ahead of server time."
- Seats:** A text input field containing the number "1".
- Buttons:** Three buttons at the bottom: "SAVE", "Save and add another", and "Save and continue editing".

Рисунок 3.4 – Додавання бронювань

Таблиця Dishes: назва страви, фото, ціна, знижка, алергени, посилання на категорії страв та опис. Нижче наведений програмний код створення класу Order.

```
class Order(models.Model):
    name = models.CharField(max_length=128, blank=True,
null=True, default=None)
    email = models.EmailField(max_length=254, blank=True,
null=True, default=None)
    phone = PhoneNumberField(blank=False, null=False)
    date = models.DateField(blank=True, null=True)
    time = models.TimeField(blank=True, null=True, default="11:00")
    booking = models.IntegerField(blank=True, null=True, default=1)
    status = models.BooleanField(blank=True, null=True,
default=False)
```

```
seats = models.IntegerField(default=1,
validators=[MinValueValidator(1), MaxValueValidator(22)])
created = models.DateTimeField(auto_now_add=True,
auto_now=False)
updated = models.DateTimeField(auto_now_add=False,
auto_now=True)

def str(self):
    return "%s %s %s %s" % (self.name, self.phone, self.date,
self.seats)

class Meta:
    verbose_name = 'Order'
    verbose_name_plural = 'Orders'
```

The screenshot shows a web form for adding a dish. The form is titled "Add Dish" and is set against a dark background. It includes the following fields:

- Name:** A text input field.
- Image:** A file upload field with a button labeled "Выбор файла" (Choose file) and a message "Не выбран ни один файл" (No file selected).
- Price:** A text input field with the value "0".
- Discount:** A text input field with the value "0".
- Alergic:** A text input field.
- Category:** A dropdown menu with a plus icon for adding new categories.
- Description:** A large text area for entering the dish's description.

Рисунок 3.5 – Добавления Страв

Форма додавання категорій до таблиці Categories зображена на рис. 3.6.

Рисунок 3.6 – Додавання Категорій

Була розроблена аутентифікація адміністратора (рис. 3.7-3.8). Нижче наведений програмний код реалізації аутентифікації.

```
<form class="form-signin" method="post" action="{% url 'login' %}">
  <h1 class="h3 mb-3 font-weight-normal">Admin Login</h1>
  {% csrf_token %}
  <input id="inputEmail" class="form-control" placeholder="Username"
required="" autofocus="" type="username" {{ form.username }}
  <input id="inputPassword" class="form-control"
placeholder="Password" required="" type="password" {{ form.password }}
  <button class="btn btn-lg btn-primary btn-block" type="submit">Sign
in</button>
  <input type="hidden" name="next" value="{{ next }}" />
</form>
```

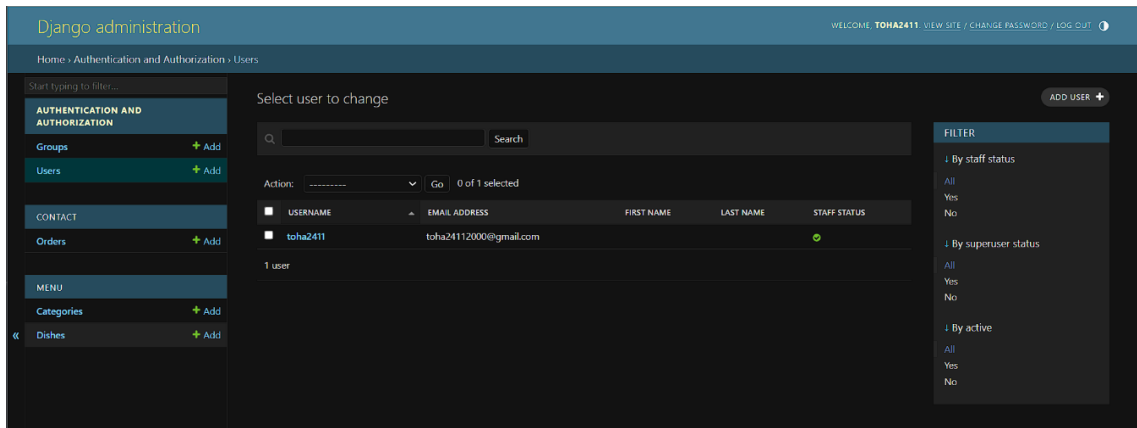


Рисунок 3.7 – Створена БД

Рисунок 3.8 – Додавання нових адміністраторів

Розроблено і реалізовано різні сторінки, такі як домашня сторінка, сторінка з формою бронювання, контактна сторінка та інформаційна сторінка.

За допомогою коду було реалізовано пункти вибору категорії меню .

```
<div class="menu-links">
```

```
<ul class="list-inline me-auto ms-auto">
```

```
<li class="list-inline-item"><a class="btn btn-theme-secondary"
onclick=" change_render(1) " target="_blank">Фастфуд</a></li>
```

```
<li class="list-inline-item"><a class="btn btn-theme-secondary"
onclick=" change_render(2) " target="_blank">Салати</a></li>
```

```
<li class="list-inline-item"><a class="btn btn-theme-secondary"
onclick=" change_render(3) " target="_blank">М'ясо</a></li>
```

```

        <li class="list-inline-item"><a class="btn btn-theme-secondary"
onclick=" change_render(4) " target="_blank">Гарнір</a></li>
        <li class="list-inline-item"><a class="btn btn-theme-secondary"
onclick=" change_render(5) " target="_blank">Напої</a></li>
    </ul>
</div>

```

Для того, щоб меню виводило окремо категорії було застосовано наступний код:

```

function change_render(a){
    if (a == 1){
        document.getElementById('view1').style.display='initial'
        document.getElementById('view2').style.display='none'
        document.getElementById('view3').style.display='none'
        document.getElementById('view4').style.display='none'
        document.getElementById('view5').style.display='none'
    }
    if (a == 2){
        document.getElementById('view1').style.display='none'
        document.getElementById('view2').style.display='initial'
        document.getElementById('view3').style.display='none'
        document.getElementById('view4').style.display='none'
        document.getElementById('view5').style.display='none'
    }
    if (a == 3){
        document.getElementById('view1').style.display='none'
        document.getElementById('view2').style.display='none'
        document.getElementById('view3').style.display='initial'
        document.getElementById('view4').style.display='none'
    }
}

```

```

    document.getElementById('view5').style.display='none'
  }
  if (a == 4){
    document.getElementById('view1').style.display='none'
    document.getElementById('view2').style.display='none'
    document.getElementById('view3').style.display='none'
    document.getElementById('view4').style.display='initial'
    document.getElementById('view5').style.display='none'
  }
  if (a == 5){
    document.getElementById('view1').style.display='none'
    document.getElementById('view2').style.display='none'
    document.getElementById('view3').style.display='none'
    document.getElementById('view4').style.display='none'
    document.getElementById('view5').style.display='initial'
  }
}

```

Реалізовано функцію у файлі views яка виводить сторінку Меню а також присвоює змінним меню з БД по категоріям.

```

def menu(request):
    products_fast_food = Dish.objects.filter(category__id=6)
    products_salat = Dish.objects.filter(category__id=7)
    products_meet = Dish.objects.filter(category__id=8)
    products_side_dishes = Dish.objects.filter(category__id=9)
    products_drinks = Dish.objects.filter(category__id=10)
    return render(request, 'Menus/Menus.html', locals())

```

Цикли виводу меню по різним категоріям реалізовано наступним кодом:


```

<div id = "view1">
  <div class="section section1">
    <div class="container">
      <div class="row">
        {% for products in products_fast_food% }
          {% include "Menus/Dish_Item.html" % }
        {% endfor % }
      </div>
    </div>
  </div>
</div>

```

Реалізовано API Google maps для відображення місцезнаходження ресторану.

Перевірка та обробка даних, отриманих від клієнта, і відповідна взаємодія з базою даних реалізовано на рисунку 3.9.

Name	Email	Phone	Seats	Date	Time	Status	Confirm	Delete
Anton Mykhilichenko	toha24112000@gmail.com	+380954109126	4	June 14, 2023	1 p.m.	False	Підтвердити	Видалити
Anton	toha24112000@gmail.com	+380954109126	19	June 8, 2023	noon	False	Підтвердити	Видалити
Ostin	toha24112000@gmail.com	+380954109126	7	July 14, 2023	7 p.m.	True	Підтвердити	Видалити

Рисунок 3.9 – Вигляд таблиці замовлень на адміністративній сторінці

Останнім кроком було протестовано весь функціонал web-додатку та не виявлено помилок під час користування (рис. 3.10).

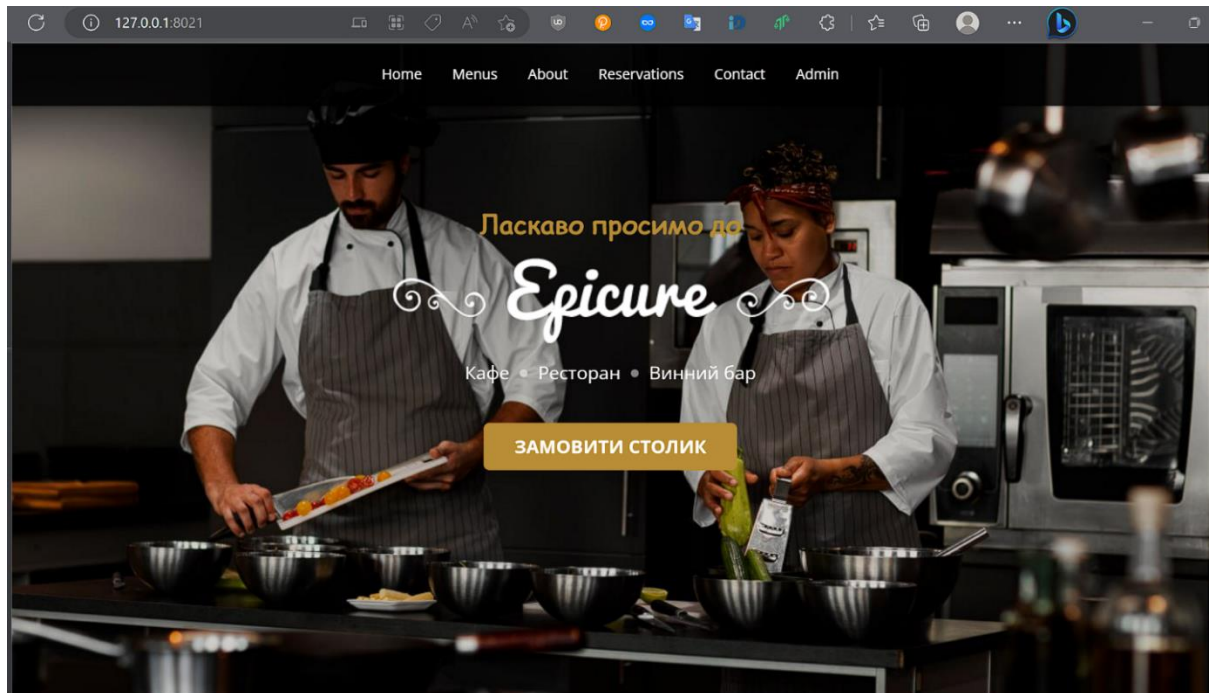


Рисунок 3.10 – Запуск додатку

3.3. Робота користувача з web-додатком

Після відкриття web-додатку можна побачити головну сторінку (рис. 3.11). Користувач бачить символіку закладу та може перейти до форми замовлення столу.

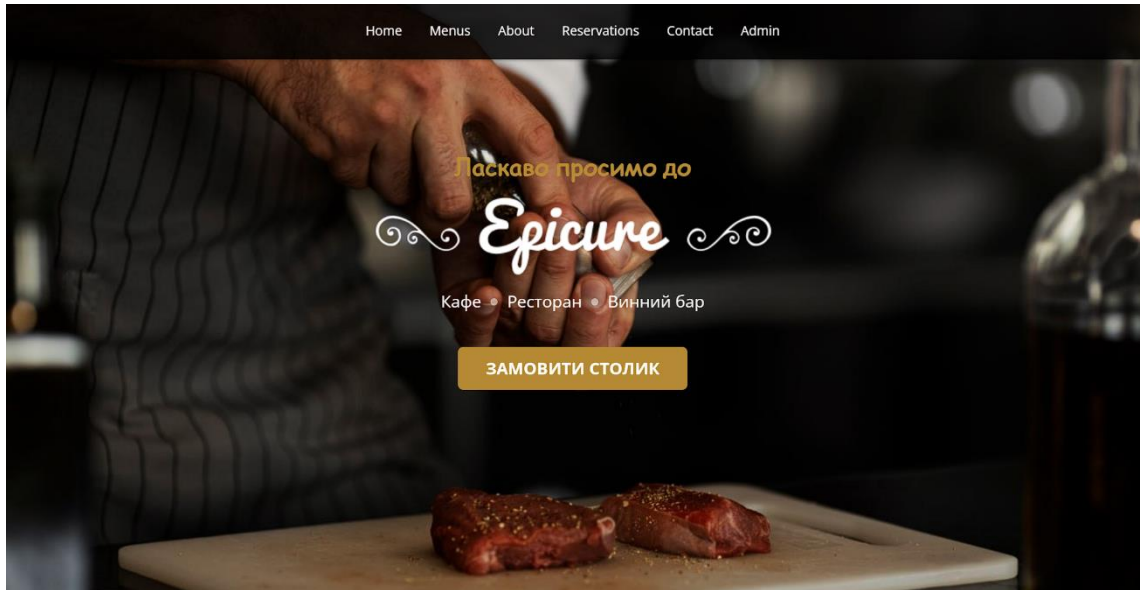


Рисунок 3.11 – Головна сторінка web-додатку

На сторінці меню розташовані категорії, які показують інформацію про страви. При відкритті можна переглянути все меню (рис. 3.12-3.13).

приготована на вогні, щоб забезпечити неперевершений смак та неповторний досвід. Розкрийте свій апетит і дозвольте нам подарувати вам незабутню подорож у світ м'ясних витонченостей.

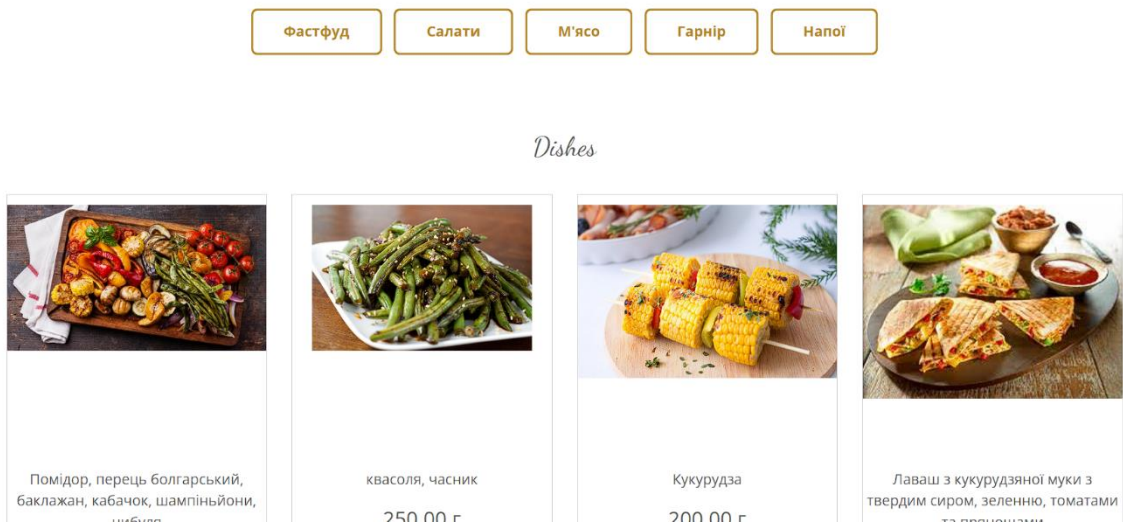


Рисунок 3.12 – Сторінка меню web-додатку

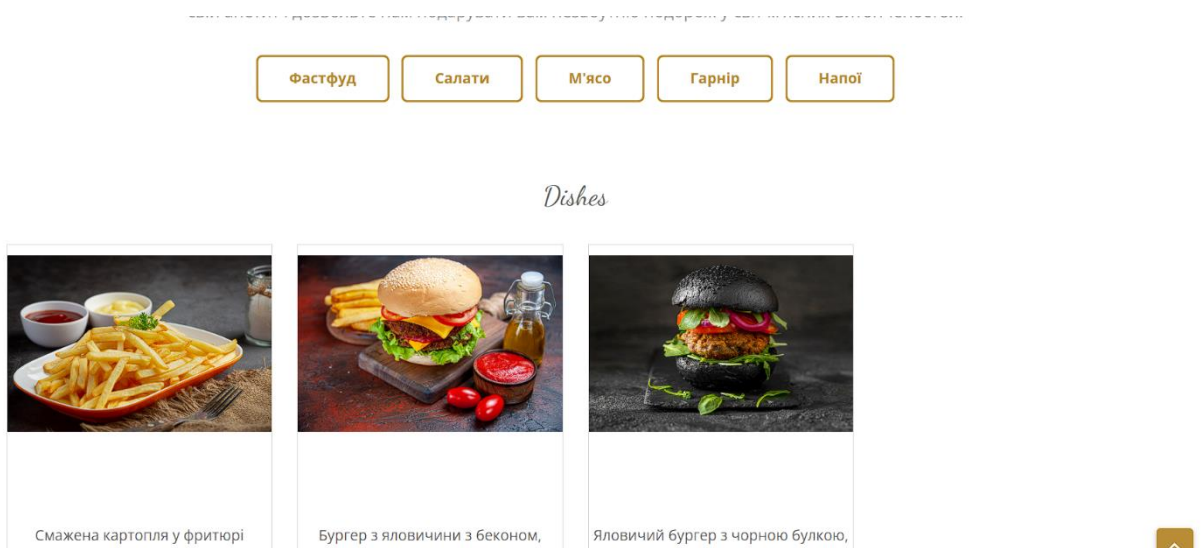


Рисунок 3.13 – Сторінка меню web-додатку

Переходячи на вкладку «Про заклад», відкривається сторінка, на якій можна переглянути інформацію про заклад (рис. 3.14).

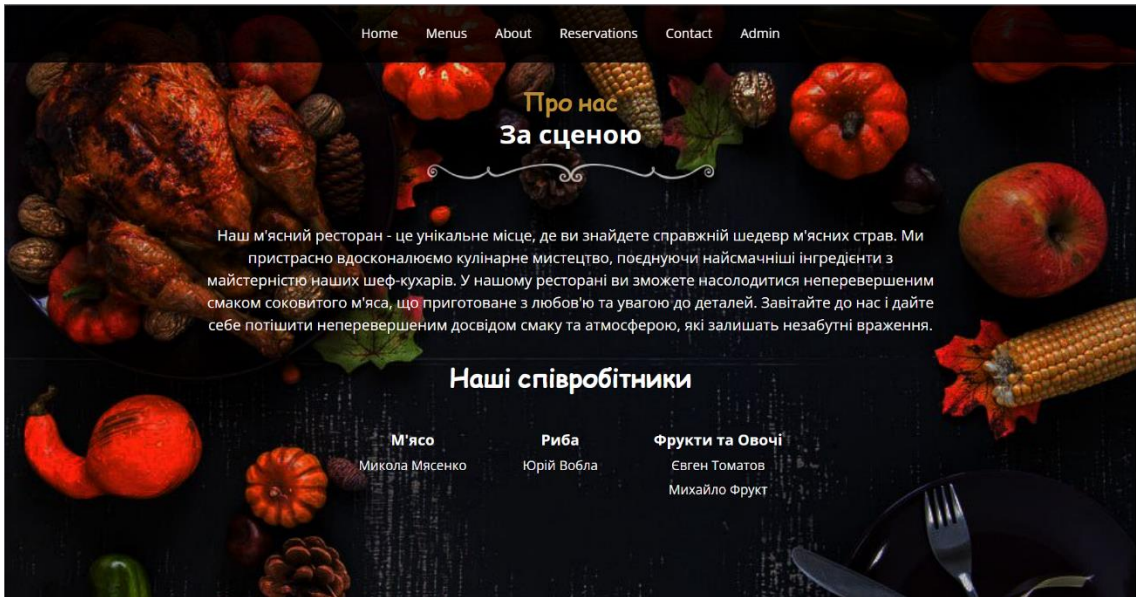


Рисунок 3.14 – Сторінка про заклад

Форма заповнення про бронювання столів містить інформацію про час, кількість відвідувачів, ім'я, електронну пошту та номер телефону (рис. 3.15).

Рисунок 3.15 – Форма бронювання

Сторінка «Контакти» містить інформацію про місцезнаходження закладу, пошту, номер телефону та кнопку про бронювання столів (рис. 3.16).

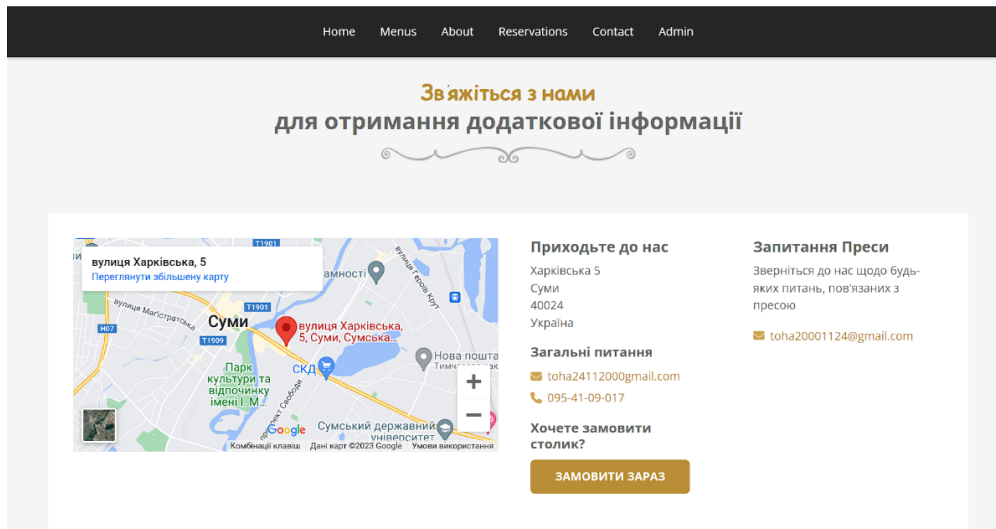


Рисунок 3.16 – Сторінка «Контакти»

3.4. Робота адміністратора з web-додатком

Адміністратор може додавати до меню страви, вказавши необхідну інформацію на сторінці (рис. 3.17-3.19).

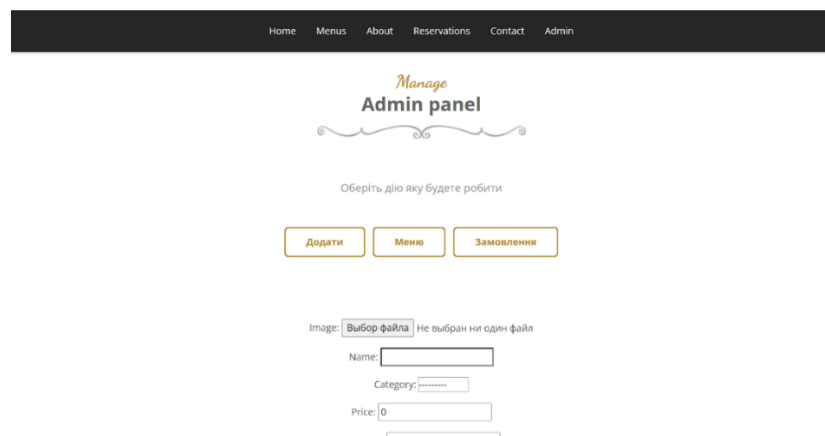


Рисунок 3.17 – Додавання страви

Home Menus About Reservations Contact Admin

Оберіть дію яку будете робити

Додати Меню Замовлення

Image: photo_2023...21-27-54.jpg

Name:

Weight:

Category:

Price:

Discount:

Allergic:

Рисунок 3.18 – Заповнена форма додавання страви

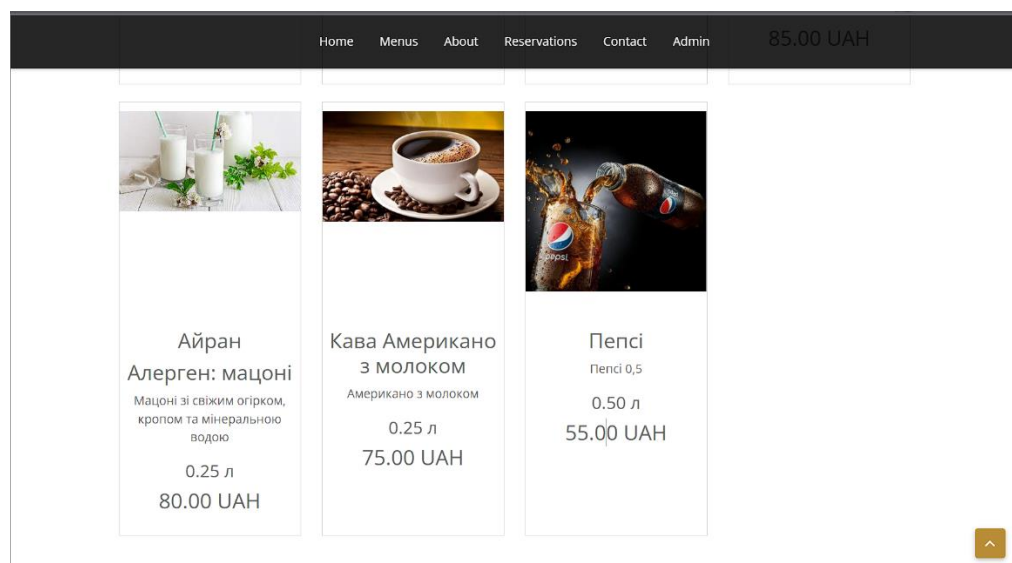


Рисунок 3.19 – Відображення нової страви

Адміністратор може переглядати інформацію, яку користувачі надіслали через форму бронювання, а також може підтвердити або видалити бронювання(рис. 3.20).

Home Menus About Reservations Contact Admin

Оберіть дію яку будете робити

Додати Меню Замовлення

Name	Email	Phone	Seats	Date	Time	Booking	Status	Confirm	Delete
Anton Mykhilichenko	toha24112000@gmail.com	+380954109126	4	June 14, 2023	1 p.m.	1 h.	False	Підтвердити	Видалити
Anton	toha24112000@gmail.com	+380954109126	19	June 8, 2023	noon	1 h.	False	Підтвердити	Видалити
Ostin	toha24112000@gmail.com	+380954109126	7	July 14, 2023	7 p.m.	1 h.	True	Підтвердити	Видалити

↑

Рисунок 3.20 – Сторінка перегляду бронювань

ВИСНОВКИ

Метою кваліфікаційної роботи бакалавра було спроектувати, розробити та впровадити web-додаток, який полегшує бронювання столиків, підвищує ефективність співпрацю з відвідувачами та покращити загальний досвід клієнтів.

Завдяки використанню різних технологій, таких як Django, SQL та JavaScript, було створено повнофункціональний web-додаток, який включає такі функції, як система онлайн-бронювання, система наповнення меню та система зворотного зв'язку з клієнтами.

Web-додаток, розроблений в рамках цього проекту, має потенціал для підвищення ефективності та прибутковості ресторану, а також покращення якості обслуговування клієнтів.

Більше того, web-додаток, розроблений в рамках цього проекту, може бути вдосконалений і розширений у майбутньому. Наприклад, можна додати додаткові функції, такі як інтеграція з соціальними мережами та програмами лояльності, щоб підвищити залученість та лояльність клієнтів. Крім того, можна було б впровадити алгоритми машинного навчання для аналізу даних про клієнтів і надання їм персоналізованих рекомендацій.

Цей дипломний проект продемонстрував потенційні переваги такого додатку як для бізнесу, так і для клієнтів, а також проблеми та рішення, які виникають при його розробці. Оскільки технології продовжують розвиватися, для ресторанів важливо адаптувати і використовувати web-додатки для поліпшення своєї роботи і обслуговування клієнтів, і цей проект служить прикладом потенціалу, який можуть запропонувати такі додатки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Jain, S., & Mishra, S. (2021). Web-based application for restaurant management: A study on its effectiveness. *International Journal of Computer Science and Information Technology*, 12(1), 14-26. https://studentnet.cs.manchester.ac.uk/resources/library/thesis_abstracts/MSc13/FullText/Tan-ChinLoong-fulltext.pdf (дата звернення: 15.05.2023).
2. Li, X., Liang, X., & Li, D. (2020). Mobile applications for restaurant management: A review. *Journal of Hospitality and Tourism Technology*, 11(4), 552-572. https://www.researchgate.net/publication/280180799_Mobile_Applications_in_the_Hospitality_Industry (дата звернення: 15.05.2023).
3. Restaurant Website Design, The Easy Way. URL: <https://restaurantengine.com/> (дата звернення: 17.05.2023).
4. Advantages of learning java script. URL: <https://formula.kr.ua/tsikavinki/perevagi-vivchennyajavascript.html> (дата звернення: 17.05.2023).
5. Advantages of learning java script. <https://formula.kr.ua/tsikavinki/perevagi-vivchennyajavascript.html>. URL: <https://formula.kr.ua/tsikavinki/perevagi-vivchennyajavascript.html> (дата звернення: 17.05.2023).
6. The Original. URL: <https://www.theogdenver.com/menus/#drinks> (дата звернення: 18.05.2023).
7. Federalist Pig. URL: <https://www.federalistpig.com/> (дата звернення: 18.05.2023).
8. Anton's URL: <https://www.antonsnyc.com/> (дата звернення: 18.05.2023).
9. Damian. URL: <https://www.damiandtla.com/> (дата звернення: 18.05.2023).

10. Python Web Development - Django Tutorial - GeeksforGeeks. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/python-web-development-django-tutorial/> (дата звернення: 31.05.2023).
11. What is IDEF - Definition, Methods, and Benefits - Edraw. [OFFICIAL] Edraw Software: Unlock Diagram Possibilities. URL: <https://www.edrawsoft.com/what-is-idef.html> (дата звернення: 23.05.2023).
12. Архітектура веб-додатків URL: <https://blog.ithillel.ua/articles/web-application-architecture> (дата звернення: 23.05.2023).
13. Persist data with SQLite. Flutter documentation | Flutter. URL: <https://docs.flutter.dev/cookbook/persistence/sqlite> (дата звернення: 23.05.2023).
14. SQLite documentation. SQLite Home Page. URL: <https://www.sqlite.org/docs.html> (дата звернення: 23.05.2023).
15. SQLite. Expo Documentation. URL: <https://docs.expo.dev/versions/latest/sdk/sqlite/> (дата звернення: 23.05.2023).
16. How to create a work breakdown structure. Lucidchart. URL: <https://www.lucidchart.com/blog/how-to-create-a-work-breakdown-structure-and-why-you-should> (дата звернення: 19.05.2023).
17. Open broadcaster software (OBS) | OBS | instructional technology tools | learning design and technology | departments | information technology | bradley university. Home | Bradley University. URL: <https://www.bradley.edu/sites/it/departments/ldt/it-tools/obs/> (дата звернення: 19.05.2023).
18. What is a gantt chart? URL: <https://www.apm.org.uk/resources/find-a-resource/gantt-chart/> (дата звернення: 18.05.2023).
19. UML use case diagram tutorial. URL: <https://www.lucidchart.com/pages/uml-use-case-diagram> (дата звернення: 19.05.2023).
20. What is activity diagram?. Ideal Modeling & Diagramming Tool for Agile Team Collaboration. URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/> (дата звернення: 20.05.2023).

ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ

ТЕХНІЧНЕ ЗАВДАННЯ на розробку інформаційної системи «Web-додаток підтримки роботи ресторану»

1. Призначення й мета створення web-додатку

1.1 Призначення web-додатку

Web-додаток призначений для підтримки роботи ресторану, а саме впорядкувати та покращити щоденні операції ресторану, надавши ефективну та зручну платформу для управління бронюваннями та наданням загальної інформації про заклад.

1.2 Мета створення web-додатку

Мета – це створення web-додатку підтримки роботи ресторану, спрощення умови бронювання столиків та надання інформації про мережу ресторанів.

1.3 Цільова аудиторія

Цільовою аудиторією є адміністратор, що керує web-додатком та має змогу редагувати певну інформацію. Також це клієнти закладу та зацікавлені відвідувачі мережі ресторанів.

2 Вимоги до web-додатку

2.1 Вимоги до web-додатку в цілому

2.1.1 Вимоги до структури й функціонування web-додатку

Web-додаток підтримки роботи ресторану повинен забезпечувати визначений набір функціональних можливостей, таких як: бронювання столиків, ознайомлення відвідувачів з закладом та меню, тощо.

Кінцевий продукт даного проекту має бути представлений web-додатком, який містить якісне інформаційне наповнення та графічні матеріали.

2.1.2 Вимоги до персоналу

Персонал ресторану не повинен мати особливих технічних навичок для роботи з web-додатком і його підтримкою. Єдиною вимогою є наявність навичок користування персональним комп'ютером та web-браузером.

2.1.3 Вимоги до збереження інформації

Уся інформація надана у web-додатку повинна зберігатися у базі даних реалізованій засобами системи управління базами даних MySQL.

2.1.4 Вимоги до розмежування доступу

Розроблюваний web-додаток має бути загальнодоступним у мережі Інтернет. Права доступу до інформації розмежовані за групами користувачів: адміністратор, відвідувач та клієнт. Адміністратор має необмежений доступ до даних з правами перегляду, додавання, редагування та видалення. Доступ до адміністративної панелі надається за спеціальним логіном та паролем.

Відвідувач web-додатку може переглядати інформацію, робити онлайн замовлення, бронювати столики та дивитися актуальну інформацію про сам заклад.

2.2 Структура web-додатку

2.2.1 Загальна інформація про структуру web-додатку

До структури web-додатку входять усі його web-сторінки, які є загальнодоступними, та адміністративна панель для наповнення web-додатку новими статтями.

Перелік сторінок web-додатку наступний:

- Головна сторінка містить переходи до всіх сторінок, з неї можна авторизуватися, зареєструватися, зробити забронювати місце в залі, та дивитися інформацію та відгуки від інших відвідувачів;

- Адміністративна сторінка містить форму додавання страв, редагування, а також видалення неактуальної інформації.

2.2.2 Навігація

Для зручної навігації повинно бути створене меню, що забезпечить швидке переміщення користувача по всім доступним сторінкам web-додатку. Меню міститиме вкладки для напоїв, перших, других страв, закусок та новинок. Меню має бути закріплене і розташовуватися зверху (у шапці) на кожній сторінці.

2.2.3 Наповнення web-додатку (контент)

Управління контентом web-додатку має здійснюватися за допомогою адміністративної панелі. Усе інформаційне наповнення web-додатку має міститися у базі даних. Графічні матеріали та інформацію для наповнення надає Замовник.

2.2.4 Дизайн та структура додатку

Дизайн web-додатку має бути виконаний у мінімалістичному та сучасному стилі. Корпоративними кольорами мережі ресторанів є палітра з теплих кольорів. Основна палітра кольорів буде містити такі чотири відтінки: F5EFE6, E8DFCA, AEBDCA, 7895B2. Тому під час розробки web-додатку треба використовувати саме ці кольори.

Види і розміри шрифтів повинні бути комфортними для перегляду. Шрифт *Vevas Neue Cyrillic* підійде для головної інформації – знижок та спеціальних пропозицій, а, округлий у порівнянні з першим, *Open Sans* стане в нагоді для оформлення уточнюючої інформації, він простий і не відволікатиме від головного. Інформаційні блоки, графічні матеріали та інші елементи web-сторінок повинні мати зручне і логічне розташування.

Розташування елементів на головній сторінці web-додатку схематично показано на рисунку А.1.



Рисунок А.1 – Схема головної сторінки

2.2.5 Система навігації (карта web-додатку)

Карта web-додатку зображена на рисунку А.2.



Рисунок А.2 – Карта web-додатку

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Потреби користувача, визначені на основі рішення замовника, представлені у таблиці А.1.

Таблиця А.1 – Потреби користувача

ID	Потреби користувача	Джерело
UN-01	Перегляд меню	Клієнт, Відвідувач
UN-02	Перегляд навантаженості закладу	Клієнт, Відвідувач
UN-03	Перегляд коментарів інших відвідувачів web-додатку, та їх емоцій після відвідування	Клієнт, Відвідувач
UN-04	Перегляд інформації про заклад	Клієнт, Відвідувач
UN-05	Редагування даних	Адміністратор
UN-06	Автоматизована побудова графіку кількості вдалих та невдалих замовлень	Адміністратор

2.3.2 Функціональні вимоги

Проаналізувавши потреби користувачів та персоналу web-додатку було визначено наступні вимоги:

- наявність реєстрації та авторизації клієнтів;
- легке спілкування з адміністраторами, швидкі відповіді та максимальне розуміння відвідувача;
- наявність панелі адміністратора для додавання, редагування та видаленні інформації з web-додатку;

2.3.3 Системні вимоги

Даний розділ визначає, розподіляє та вказує на системні вимоги, визначені розробником. Їх перелік наведений в таблиці А.2.

Таблиця А.2 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
SR-01	Каталог меню	M	Формує можливість перегляду
SR-02	Каталог послуг	S	Формує можливість перегляду послуг
SR-03	Таблиця з бази даних із замовленнями	M	Надає можливість відображати заброньовані послуги
SR-04	Таблиця з бази даних з контентом	M	Відповідає за заповнення web-додатку контентом
SR-05	Таблиця з бази даних з цінами на послуги	M	Надає можливість подальшого відображення цін на всі види послуг
SR-06	Наявність модуля зворотного зв'язку	S	Надає можливість клієнту зв'язатися з адміністратором закладу
SR-07	Панель адміністратора	M	Відповідає за подальше наповнення та редагування контенту адміністратором

Умовні позначення в таблиці А.2:

Must have (M) – вимоги, які повинні бути реалізовані в системі;

Should have (S) – вимоги, які мають бути виконані, але вони можуть почекати своєї черги;

Could have (C) – вимоги, які можуть бути реалізовані, але вони не є центральною ціллю проекту.

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

Реалізація web-додатку відбувається з використанням:

- WordPress
- Python
- MySQL 8.0

2.4.2 Вимоги до лінгвістичного забезпечення

Web-додаток має бути виконаний українською мовою.

2.4.3 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Web-браузер: Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище.

3 Склад і зміст робіт зі створення web-додатку

Докладний опис етапів роботи зі створення web-додатку наведено в таблиці А.3.

Таблиця А.3 – Етапи створення web-додатку

№	Назва роботи проекту	Тривалість
1	Розробка web-додатку ресторану	142 днів
1.1	Дослідження предметної області	15 днів
1.1.1	Аналіз предметної області	8 днів
1.1.2	Аналіз існуючих програмних продуктів для вирішення поставленої задачі	7 днів

Продовження таблиці А.3

1.2	Постановка задачі	18 днів
1.2.1	Формування переліку вимог до програми	13 днів

1.2.2	Вибір засобів реалізації	5 днів
1.3	Моделювання програмного продукту	32 дні
1.3.1	Моделювання програмного продукту в нотації IDEF0	7 днів
1.3.2	Розробка моделі аналізу	7 днів
1.3.3	Створення моделі проектування	10 днів
1.3.4	Розробка моделі реалізації	8 днів
1.4	Реалізація програмного продукту	37днів
1.4.1	Верстка сторінок	17 днів
1.4.2	Робота з бек-енд	10 днів
1.4.3	Розробка і впровадження бази даних	10 днів
1.5	Тестування програмного продукту	12 днів
1.5.1	Beta-тестування	7 днів
1.5.2	Alpha-тестування	5 днів
1.6	Публікація на хостинг	1 день
1.7	Написання програмної документації	10 днів

3 Вимоги до складу й змісту робіт із введення web-додатку в експлуатацію

Web додаток має бути затверджений та розміщений на хостингу, тому необхідно придбати доменне ім'я та місце на хостингу. На хостинг переноситься web-додаток і наповнення бази даних з подальшою їх доробкою.

ДОДАТОК Б. ПЛАНУВАННЯ РОБІТ

Успіх будь-якого проекту значною мірою залежить від ефективного планування та управління. Особливо це стосується проектів з розробки програмного забезпечення, таких як розробка web-додатку для підтримки роботи ресторану. Для того, щоб забезпечити своєчасне та успішне завершення проекту, детальний робочий план має вирішальне значення.

У цьому підрозділі буде описано робочий план розробки web-додатку для ресторану, який включає в себе розбивку проекту на керовані завдання, планування цих завдань і розподіл ресурсів для їх виконання. У робочому плані буде використана структура розбиття робіт (WBS) і діаграма Ганта, щоб забезпечити чітке візуальне представлення графіка проекту і взаємозалежностей між завданнями.

WBS розділить проект на менші, більш керовані завдання, які будуть призначені конкретним членам команди [16]. Кожне завдання матиме чіткий опис його результатів і залежностей від інших завдань, що полегшить членам команди розуміння своїх обов'язків і термінів виконання проекту. Діаграма Ганта надасть часовий графік проекту, включаючи дати початку і завершення кожного завдання, а також критичний шлях проекту.

Загалом, робочий план слугуватиме дорожньою картою для команди розробників, забезпечуючи чіткий напрямок і графік завершення проекту. У наступних розділах цього підрозділу будуть описані конкретні завдання та графік розробки web-додатку.

Деталізація мети проекту методом SMART. Деталізація SMART-цілей - це метод, який використовується для створення конкретних, вимірюваних, досяжних, релевантних і обмежених у часі (SMART) цілей проекту. Цей процес передбачає розбиття великої мети на менші, більш конкретні цілі, які відповідають критеріям SMART.

Кожен елемент абрєвіатури SMART представляє окремий аспект цілі, який необхідно визначити. Наприклад, мета повинна бути конкретною, тобто чітко визначеною і зрозумілою для всіх учасників процесу. Вимірюваність означає, що мета повинна мати кількісний вимірник або індикатор для визначення того, чи була вона досягнута. Досяжність означає, що мета має бути реалістичною з урахуванням наявних ресурсів і можливостей. Релевантність означає, що мета повинна відповідати загальним цілям і місії проекту. Нарешті, обмеженість у часі означає, що мета повинна мати визначений кінцевий термін або графік для завершення. Дотримуючись процесу деталізації цілей SMART, керівники проектів можуть створювати чіткі та досяжні цілі, які відповідають загальним цілям та місії проекту. Це допомагає гарантувати, що проект не відстає від графіка і що всі учасники розуміють свою роль і обов'язки в досягненні цілей проекту. Результати деталізації методом SMART наведено в таблиці Б.1.

Таблиця Б.1 – Деталізація мети проекту методом SMART

Specific	Створити багатофункціональний web-додаток підтримки діяльності мережі ресторанів із метою поширення та збільшення продажів продукції серед користувачів мережі Інтернет в Україні
Measurable	У середньому за місяць web-додаток буде здатний проводити близько 50 онлайн замовлень з усіх наявних категорій
Achievable	Є затверджене технічне завдання проекту, маємо необхідне обладнання для створення web-додатку, кваліфікований склад робітників різних напрямлень для створення і покращення функціонування web-додатку
Relevant	Метою створення є забезпечення якісного обслуговування клієнтів, надання якісної та доступної продукції; підвищення конкурентоспроможності серед інших ресторанів
Time-framed	Є конкретний термін – 06 червня 2023 р.

На рисунку Б.2 представлено організаційну структуру планування проекту. Список виконавців, які функціонують у даному проекті, описано в таблиці Б.1.



Рисунок Б.1 – WBS-структура робіт проекту

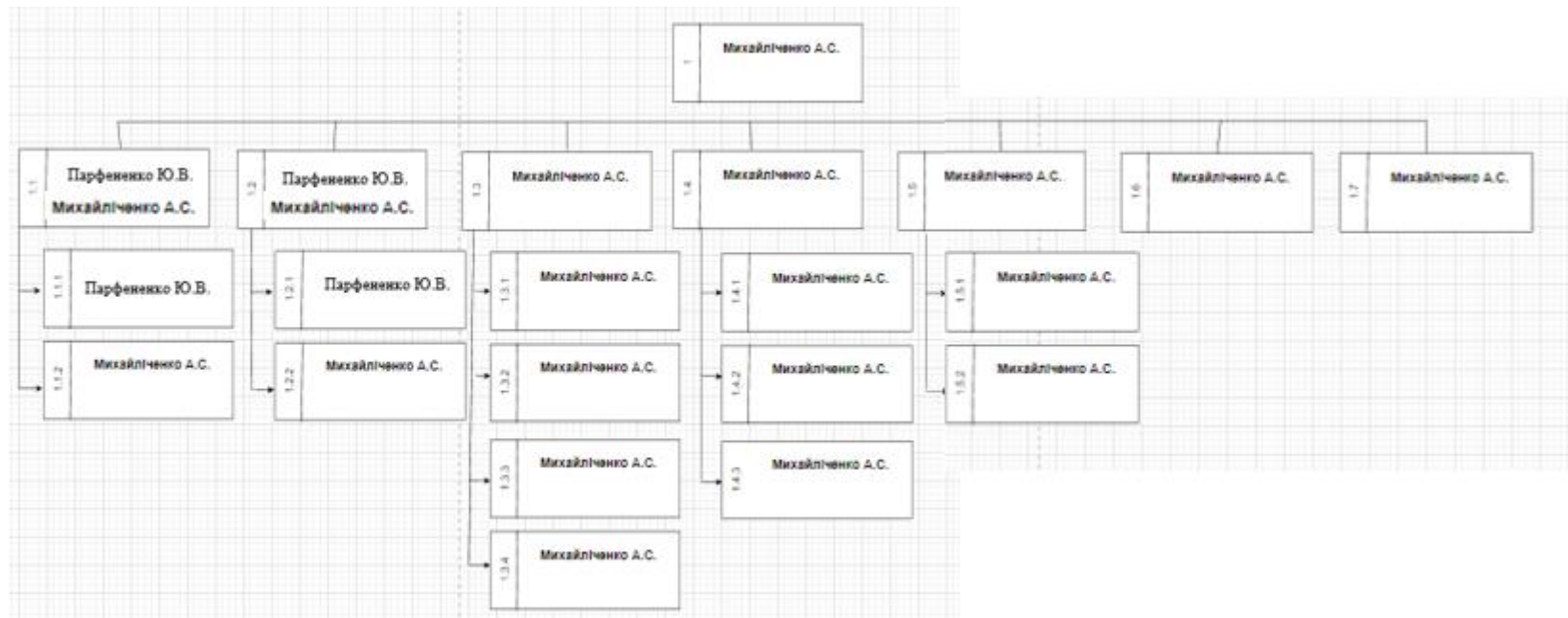


Рисунок Б.2 – OBS-структура робіт проект

Таблиця Б.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Михайліченко А.С.	Створює шаблони web-додатку, виконує front-end та back-end розробку
Проектувальник	Михайліченко А.С.	Розробляє структуру web-додатку
Тестувальник	Парфененко Ю.В. Михайліченко А.С.	Відповідають за тестування функціоналу та дизайну web-додатку
Керівник проекту	Парфененко Ю.В.	Формує завдання на розробку проекту, проводить консультації
Менеджер проекту	Михайліченко А.С.	Відповідає за виконання термінів, розподіл ресурсів та завдань між учасниками. Відповідальний за супровід проекту документацією

Діаграма Ганта. Діаграма Ганта – це тип гістограми, який ілюструє графік проекту. Вона відображає горизонтальну часову шкалу з кожним завданням, представленим у вигляді відрізка, який відповідає тривалості завдання. Дати початку і закінчення кожного завдання позначені на часовій шкалі, а відрізки можуть мати колірне кодування для позначення статусу кожного завдання [18].

Діаграми Ганта – це популярний інструмент, який використовується менеджерами проектів для планування, складання розкладу та відстеження прогресу проекту. Вони забезпечують візуальне представлення розкладу проекту, включаючи залежності між завданнями та аналіз критичного шляху.

Це полегшує менеджерам проєктів виявлення потенційних конфліктів у плануванні та відповідне коригування плану проєкту.

Діаграми Ганта можна використовувати для найрізноманітніших проєктів, від простих до складних. Вони особливо корисні для проєктів, які мають багато взаємозалежних завдань або вимагають координації між кількома членами команди. Використовуючи діаграму Ганта, менеджери проєктів можуть відстежувати хід виконання проєкту і гарантувати, що він не відстає від графіка і дотримується встановлених термінів.

Календарний графік даного проєкту представлено на рисунках Б.3-Б.4.

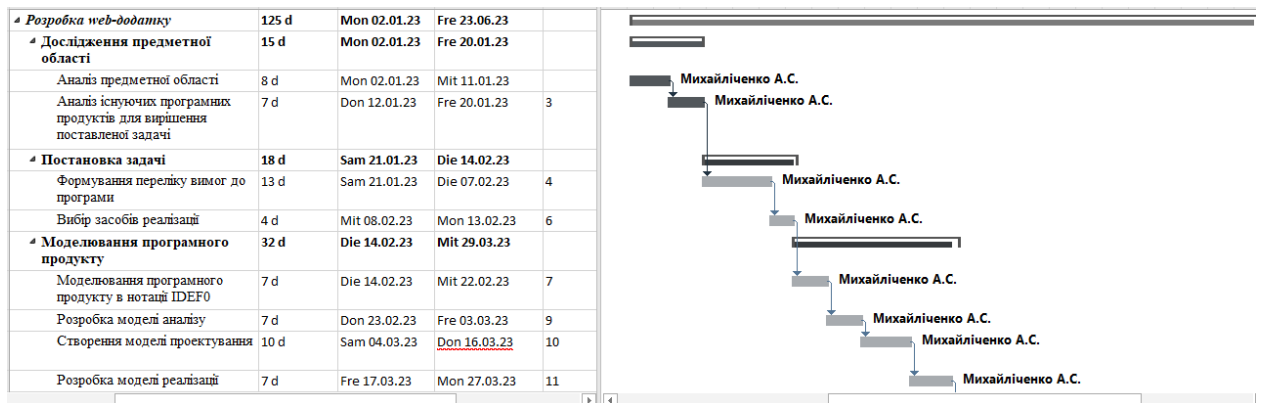


Рисунок Б.3 – Календарний графік проєкту

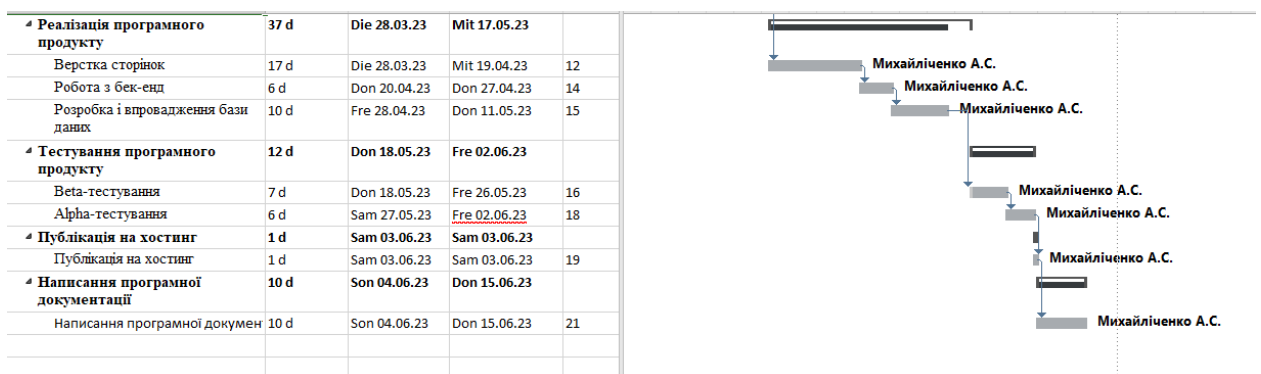


Рисунок Б.4 – Продовження календарного графіку проєкту

Управління ризиками проєкту. Управління проєктними ризиками – це процес виявлення, оцінки та управління потенційними ризиками, які можуть вплинути на досягнення цілей проєкту. Метою управління ризиками є

мінімізація ймовірності та впливу негативних подій і максимізація ймовірності та впливу позитивних подій.

Ефективне управління проектними ризиками вимагає постійного моніторингу та аналізу протягом усього життєвого циклу проекту. Це дозволяє керівникам проектів виявляти нові ризики, які можуть виникнути, і відповідно коригувати свої стратегії. Завдяки ефективному управлінню проектними ризиками керівники проектів можуть мінімізувати потенційні негативні наслідки та підвищити загальний успіх проекту.

У таблиці Б.3 представлено шкалу для класифікації ризиків за величиною впливу на проект та ймовірністю виникнення.

Таблиця Б.3 – Шкала оцінювання ризиків за ймовірністю виникнення та величиною впливу

Оцінка	Ймовірність виникнення	Вплив ризику	Тип ризику
1	Низька	Низький	Прийнятний
2	Середня	Середній	Виправданий
3	Висока	Високий	Недопустимий

Визначені ризики для даного проекту наведено в таблиці Б.4. У результаті планування реагування було отримано матрицю ймовірності виникнення ризиків та їх впливу. Результати занесено в таблицю Б.5. Зеленим кольором позначено прийнятні ризики, жовтим – виправдані, а червоним – недопустимі.

Таблиця Б.4 – Ідентифікація ризиків

№	Назва (опис) ризику
1	Поява альтернативного продукту
2	Затримки у зв'язку з відключенням електроенергії

3	Низька кваліфікація розробників
4	Неоптимальний розподіл часу
5	Часте внесення змін у ТЗ
6	Повітряні тривоги/обстріли
7	Відсутність/закінчення ліцензії ПЗ
8	Вигорання розробника

Таблиця Б.5 – Оцінка ймовірності, впливу та рангу ризиків

№ ризику	Назва (опис) ризику	Ймовірність (0,1-0,9)	Вплив (0,05-0,8)	Ранг
1	Поява альтернативного продукту	0,3	0,4	0,12
2	Затримки у зв'язку з відключенням електроенергії	0,5	0,4	0,2
3	Низька кваліфікація розробників	0,1	0,2	0,03
4	Неоптимальний розподіл часу	0,3	0,4	0,12
5	Часте внесення змін у ТЗ	0,3	0,2	0,06
6	Повітряні тривоги/обстріли	0,3	0,4	0,12
7	Відсутність/закінчення ліцензії ПЗ	0,1	0,8	0,08
8	Вигорання розробника	0,3	0,4	0,12

Таблиця Б.5 – Матриця ймовірності та впливу ризиків на проект

Ймовірність ризику (Й)	Вплив загрози (ризик)				
	Дуже малий	Малий	Середній	Великий	Дуже великий
	0,05	0,1	0,2	0,4	0,8
0,9					
0,7					
0,5				R2(0.2)	
0,3			R5(0.06)	R1(0.12) R4(0.12) R6(0.12) R8(0.12)	
0,1			R3(0.03)		R7(0.08)

Класифікація ризиків за рівнем, відповідно до отриманого значення індексу, представлена в таблиці Б.6. У таблиці Б.7 описано ризики проекту та стратегії реагування на кожен з них.

Таблиця Б.6 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять (номера)
1	Прийнятні	$0,005 \leq R \leq 0,0$ 5	3
2	Виправдані	$0,05 < R \leq 0,14$	1, 4, 5, 6, 8
3	Недопустимі	$0,14 < R \leq 0,72$	2, 7

Таблиця Б.7 – Ризики та стратегії реагування

ID ризику	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А (заходи запобігання виникненню ризику)	Тип стратегії реагування	План Б (заходи усунення наслідків ризику)
RS_1	Відкритий	Поява альтернативного продукту	Низька	Високий	0,12	1.Провести попереднє дослідження альтернативних продуктів.	Прийняття	
RS_2	Відкритий	Затримки у зв'язку з відключенням електроенергії	Середня	Високий	0,2	1.Додати до плану резервні дні. 2.Придбати додаткові павербанки. 3.Виконувати	Попередження	При виявленні даних про відключення, змінити час роботи на зручний.

						роботу на ноутбуці.		
RS _3	Від критий	Низька кваліфікація розробників	Низька	Середній	0,03	1.Орієнтуватися на пошук робітників із належною кваліфікацією до подібних проектів.	Попередження	Перед початком роботи перевіряємо кваліфікацію працівника
RS _4	Від критий	Неоптимальний розподіл часу	Середня	Високий	0,12	1.Розрахувати час виконання замовлення із включен	Попередження	Розподілити завдання на кожного робітника із чітким,

						ням резервів.		визначеним терміном часу.
RS _5	Від критий	Часте внесення змін у ТЗ	Низька	Середня	0,06	Розробники повинні обговорити ТЗ із замовником, правки будуть в будь-якому випадку, все залежить від кваліфікації розробників.	Прийняття	
RS _6	Від критий	Повітряні тривоги \ обстріли	Низька	Високий	0,12	Допомогти учасникам проекту покинути	Попередження	Слідкувати за строками виконання робіт

						обстрілювану територію.		та по можливості коригувати зв'язку по ситуації
RS_7	Відкритий	Відсутність/закінчення ліцензії ПЗ	Низька	Середня	0,03	Створювати комфортні температурні умови праці	Прийняття	
RS_8	Відкритий	Вигорання розробника	Середня	Високий	0,12	Створювати комфортні температурні умови праці	Прийняття	Надати довготривалу відпустку та (або) знайти нового розробника.