

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Створення веб-сайту для ресторану «Рандеву» на CMS платформі
Wordpress 5.8.4 з розробленням компонента для замовлення страв

Виконав: студент IV курсу, групи СН-42

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

(підпис)

Олексяк В.Д.

(прізвище та ініціали)

Керівник

(підпис)

Млинко Б.Б.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Г.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Тернопіль
2022

АНОТАЦІЯ

Створення веб-сайту для ресторану «Рандеву» на CMS платформі Wordpress 5.8.4 з розробленням компонента для замовлення страв // Кваліфікаційна робота освітнього рівня «Бакалавр» // Олексяк Віталій Дмитрович// Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНс-42 // Тернопіль, 2022 // С. 54 , рис. – 22, табл. – 0 , кресл. – 0 , додат. – 5 , бібліогр. – 30.

Ключові слова: сайт, бот, плагін, CMS, Wordpress, Web, HTML, SEO, CSS, WWW.

Кваліфікаційна робота присвячена створенню сайту «Рандеву» та телеграм-бота до нього.

Мета роботи: дослідити основні принципи створення веб-сайту за допомогою CMS платформи Wordpress 5.8.4, та розробити компонент для замовлення страв, який буде задовольняти вимоги користувачів та замовників.

В першому розділі кваліфікаційної роботи розглянуто аналіз відомих рішень за темою та виділено основні елементи які допоможуть досягти мети.

В другому розділі кваліфікаційної роботи розглянуто створення сайту, його налаштування, розроблено телеграм-бота, проведено його тестування і завантажено на сервер.

В третьому розділі кваліфікаційної роботи розглянуто питання долікарської допомога при харчових отруєннях та протипожежних вимоги до виробничого освітлення.

ANNOTATION

Creating a website for the restaurant "Rendezvous" on the platform CMS Wordpress 5.8.4 with a developed component for ordering dishes // Qualification work of the educational level "Bachelor" // Oleksyak Vitaliy Dmitrovich // Ternopil National Technical University named after Ivan Pulyuy, computer - of information systems and software engineering, Department of Computer Science, SNA-42 group // Ternopil, 2022 // C. 54, fig. - 22, table. - 0, chair. - 0, add. - 5, bibliogr. - 30.

Keywords: site, bot, CMS, Wordpress, Web, HTML, SEO, CSS, WWW.

Qualification work is devoted to the creation of the site "Rendezvous" and a telegram-bot to him.

Objective: to explore the basic principles of creating a website using the CMS platform Wordpress 5.8.4, and to develop a component for ordering dishes that will meet the requirements of users and customers.

The first section of the qualification work considers the analysis of known solutions on the topic and highlights the main elements that will help achieve the goal.

The second section of the qualification work considers the creation of the site, its settings, developed a telegram bot, tested it and uploaded it to the server.

The third section of the qualification work considers the issue of pre-medical care for food poisoning and fire safety requirements for industrial lighting.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

WWW – World Wide Web – Всесвітня мережа.

HTML – Hypertext Markup Language – Мова розмітки гіпертекстових документів.

HTTP – протокол передачі гіпертексту.

CSS – Cascading Style Sheets – Каскадні таблиці стилів.

DNS – Domain Name System – Доменна система імен.

SSL – криптографічний протокол, який забезпечує встановлення безпечного з'єднання між клієнтом і сервером.

CMS – система управління вмістом.

БД – База Даних.

PHP – скриптова мова програмування, була створена для генерації HTML-сторінок на стороні вебсервера.

Бот – робот, який може виконувати роботу за певним алгоритмом.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Сайт та його особливості	10
1.2 Класифікація сайтів	11
1.3 Етапи та структура створення веб-сайтів.....	15
1.4 Платформа для швидкої розробки сайту.....	17
1.5 Застосування ботів в діяльності людини.....	19
1.6 Мова програмування python, як база для розробки ботів	20
1.7 Постановка задачі на розробку.....	21
1.8 Висновки до першого розділу	21
РОЗДІЛ 2. СТВОРЕННЯ ВЕБ-САЙТУ «РАНДЕВУ» НА CMS ПЛАТФОРМИ WORDPRESS	22
2.1 Реєстрація домену	22
2.2 Налаштування сайту.....	24
2.3 Розробка телеграм бота.....	27
2.3.1 Написання коду для телеграм-бота	28
2.3.2 Тестування	28
2.3.3 Зазгужка бота на сервер.....	31
2.4 Висновки до другого розділу.....	31
РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	32
3.1 Долікарська допомога при харчових отруєннях	32
3.2 Протипожежні вимоги до виробничого освітлення.....	33
3.3 Висновки до третього розділу	35
ВИСНОВКИ.....	36
ПЕРЕЛІК ДЖЕРЕЛ	37
ДОДАТКИ	

ВСТУП

Актуальність теми. Сьогодні в інтернеті можна зустріти величезну кількість прєктів різної тематики, це різноманітні ресурси та інтернет-магазини, особисті блоги та соціальні мережі, великі портали.

Ті, хто створює та розкручує свій сайт, робить це для того, щоб на ньому заробити. Способів заробити на сайті є безліч: це і банерна реклама, тизери, і продаж посилань, і партнерські програми.

В нашій час великої популярності набувають невеликі сайти-візитки окремих організацій, інформаційні сайти персон або закладів. Наявність власного сайту відкриває для всіх видів людської діяльності чудову можливість донесення до віддалених користувачів про послуги, що пропонує заклад.

У даній роботі описано про створення веб-сайту для ресторану, на CMS платформі Wordpress 5.8.4 з розробленням компонента для замовлення страв, бронювання столика, доставка і підтримка.

Перший сайт в світі створили в 1991 році. Тім Бернерс-Лі був автором сайту. Інформація яка була опублікована на цьому сайті – опис нової технології World Wide Web. На сьогодні сайт відіграє важливу роль для компаній, фірм, інтернет-магазинів та інше. За допомогою сайту можна поширювати інформацію, швидко її коригувати, таким чином люди, які відвідали ваш сайт будуть володіти найостаннішими даними. Завдяки сайту організація має можливість донести величезну кількість інформації до людей. Жоден інший мас-медіа не здатний дати такого приголомшуючого ефекту. Тільки маючи власний сайт організація не обмежена і може розмістити будь-яку бажану кількість інформації. Сайт же доступний 24 години на добу, 7 днів на тиждень, тому всі бажаючі можуть отримати до нього доступ.

Мета і задачі дослідження. Мета даної кваліфікаційної роботи освітнього рівня «Бакалавр»: дослідити основні принципи створення веб-сайту за допомогою CMS платформи Wordpress 5.8.4, та розробити компонент для замовлення страв, який буде задовольняти вимоги користувачів та замовників.

Задачі дослідження:

- провести аналітичний огляд предметної області;
- проаналізувати засоби створення сайтів та телеграм-бота;
- дослідити і проаналізувати роботу CMS платформи WordPress;
- спроектувати структуру сайту;
- розробити веб додаток «Рандеву»;
- розробити телеграм-бота;
- провести тестування розробленого продукту.

Практичне значення одержаних результатів. Сайт «Рандеву», телеграм-бот призначений для замовлення страв, бронювання столика, доставка і підтримка.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Сайт та його особливості

На сьогоднішній день кожна людина або організація може мати свій сайт або свою «сторінку в інтернеті» для організації це може бути окремий веб-сайт, а для людини – сторінка у складі іншого сайту. Є безліч визначень що собою являє сайт найголовніші із понять:

Веб-сайт – це сукупність веб-сторінок, розташованих в мережі інтернет, власником яких є приватна особа або організація.

Сайт – це пов’язана певна кількість сторінок і пов’язаного з ними вмісту, що ідентифікується загальним доменним іменем і публікується принаймні на одному веб-сервері.

Сторінки на сайті пов’язані загальним доменним ім’ям, темою та зовнішнім виглядом. Логічно пов’язана сукупність програмного забезпечення, інформації та мультимедійних засобів. В основному, веб-сайт відображає успіх компанії, її обличчя. Сторінка результатів пошуку Google, сторінка відео YouTube і ваша сторінка у Facebook є частиною однойменних сайтів [23].

Створення сторінки сайту відбувається за допомогою мови HTML. Принцип їх роботи полягає у виведенні зображення на екран, що спостерігає користувач при введенні пошукового запиту у браузері. За допомогою мови HTML можна легко видозмінювати текст використовуючи різні функції та гіперпосилання, додавати на сторінку зображення та інші медіа-файли. Зовнішній вигляд сторінки можна змінити завдяки таблиць стилів на мові CSS. Серед функцій мови CSS можна відокремити: форматування тексту на веб-сайті, додавання нових можливостей мовою JavaScript, що дає можливість детально розглянути властивості сторінки. На сайті сторінки можуть бути як простим статичним набором файлів так і створюватись за допомогою спеціальної комп’ютерної програми на сервері. Цю програму можна реалізувати на замовлення для особистого веб-простору, або бути готовим продуктом,

передбачений на певний тип сайтів. За їх допомогою власник сайту має можливість гнучкої настройки і додавання даних на веб-сайті. Програми такого типу називають системи управління вмісту (CMS). Схожі поняття веб-сайту були вірними, коли мережа інтернету тільки розвивалась і в більшості випадків була призначена для розваг. Щоб створити такий веб-сайт необхідні лише знання мови гіпертекстової розмітки HTML. На рахунок зручності і вигляд тогочасного веб-сайту особливо не доводилось говорити.

З часом мови програмування розвиваються, інтернет мережа стає самодостатньою галуззю економіки і мільйони людей йдуть на роботу в мережу інтернет.

Основні завдання які виконує веб-сайт:

– Реклама продукції, послуги, ідеї. Правильно зроблений веб-сайт швидко переконає клієнта придбати товар, або послуг, або ідеї, що пропагуються на ньому.

– Продаж товарів, послуг, інформації, ідей. Можливість замовлення товарів і послуг, не відходячи від комп'ютера, значно розширює можливості і клієнта, і продавця.

– Безкоштовне надання інформації або послуг. Насправді надання інформації або послуг - це засіб залучення відвідувачів до даного ресурсу для здобуття, наприклад: показу реклами, якщо це рекламний майданчик.

– Підтримка клієнтів.

У більшості випадків в інтернеті одному веб-сайту відповідає одне доменне ім'я. Саме за доменними іменами сайти ідентифікуються в глобальній мережі. Можливі інші варіанти: один сайт на декількох доменах або кілька сайтів під одним доменом. За звичай декілька доменів використовують великі сайти[8].

1.2 Класифікація сайтів

Найбільш поширені веб-сайти є для великих компаній з широким спектром продажу товарів та надання послуг. Веб-сайт слугує рекламою для багатьох

компаній та фірм, виконує безліч функцій, серед яких є послуги, обслуговування клієнтів, комунікація з ними та інше.

Сайт допоможе покращити репутацію фірм, завдяки йому можна привабити нових клієнтів, збільшити попит на продажі. Сайти мають декілька критерій, серед основних є сайт вільного доступу та приватного користування. У вільному доступі використовується сайт для подачі нової інформації, продажу товарів, взаємозв'язок між користувачами завдяки онлайн сервісам.

Сайти класифікуються на два типи:

- з ціллю отримання прибутку, так звані комерційні;
- сайт в яких є подача інформації та комунікація з клієнтом, яка є не зв'язана з отриманням грошей, їх називають некомерційні.

Особливістю комерційних сайтів є розробка структури та зручний інтерфейс для користувача, мета якого є приваблення аудиторії, визначення тематики, опублікування нової інформації чи прості, як сайт візитка.

Ресурс веб сторінки на пряму залежить від його фізичного розташування. Недоліком локальних сайтів є функціонування їх в обмеженому радіусі дії. Перевагою можна вважати загальнодоступні або з відкритим середовищем для усіх бажаючих відвідувачів сайту.

Є ряд сайтів які обмежують користувачам доступ для отримання інформації, з його впливає що сайти можна поділити по доступності:

- відкриті – будь-яка інформація відкрита для всіх користувачів інтернету;
- напівзакриті – в основному це сайти на яких потрібно пройти реєстрацію, щоб написати власну статтю або виконати якусь дію на сайті;
- закриті – веб-сторінки доступ яких доступний певному колу персон (поліції, військовим і секретним службам).

У важкості оформленні сайту можна виділити три основні типи:

- статичні – побудовані на мові Html, користувач може їх отримати лише у тому вигляді, як він збережений в інтернет середовищі;

- динамічні – для роботи використовують систему управління сайтом (CMS). Зміст створюється з використанням різного типу скриптів та БД;

- флеш-сайт – основною метою даного типу є створення яскравого, запам'ятовуючого сайту з анімаційними ефектами, вмістом якого можуть бути візитки, реклами, корисна інформація тощо.

Сфера застосування сайтів є дуже широка. Кожен з типів сайту має свої властивості, особливості та відмінності. Що робить кожен веб сторінку унікальною. Якщо заглиблюватись детальніше можна сказати, що кожен тип сайтів поділяється на підкатегорії:

- інтернет магазини;
- соціальні ресурси;
- комерційні сайти;
- сайти корпоративного напрямку;
- інформаційні сайти [27].

Найбільш поширеними сайтами в мережі Інтернет є інформаційні або новинні, сайти корпорацій та масштабних організацій та онлайн магазини. Згідно статистики, яка була проведена у 2020 році проаналізовано, що середній вік відвідувачів таких веб сторінок 16-24 роки. Це свідчить про те, що найбільша активність є серед учнів та студентів. Інформаційний сайт допомагає отримати нову інформацію, завдяки пошуку в браузері в мережі можна легко отримати результат веб сторінок, на яких опублікована інформація по запиту. Приклад інформаційного сайту (див. рисунок 1.1) Сайти інформаційного характеру містять статті, наукові дослідження, наповненість самих відбувається за допомогою авторів чи власників даних веб сторінок. Інформаційні сайти можуть бути не лише джерелом нової інформації. А й стати посередником для інших сайтів, в передачі даних, обміну між іншими сайтами. Сайти інформації також містять характер новин. Дані, які розміщені на сайті новин може бути опублікована, як журналістами, так і звичайними користувачами, які вирішили поділитись подіями, що недавно відбулись. Сайти новин не завжди містять інформацію правдиву, тому варто її перевіряти на офіційних джерелах. Та

представлення фактів, які засвідчують її. Є частина користувачів, які публікують дані не правдиві, або прикрашають їх, для зацікавлення аудиторії, що в свою чергу приносить грошову винагороду, не зовсім чесним способом.



Рисунок 1.1 – Приклад інформаційного сайту

Ще одним поширеним підвидом сайту є корпоративного напрямку, до них належать веб сторінки організацій, фірм, компаній. Для усіх них сайт відіграє важливу роль у наданні послуг, зацікавлення та захоплення більшого кола клієнтів, які б скористались послугами. На таких сайтах розміщено не лише послуги, а й велика кількість реклами. Реклама у поєднанні з веб сторінками відіграє важливу роль. Реклама допомагає не лише привабити клієнтів, а й в деяких випадках покращує оформлення сторінки. Розміщення реклами на сайті інших компаній, які спільно працюють з іншими організаціями створює міцний зв'язок між ними. Приносить дохід власникам сайту. В час стрімкого розвитку інноваційних технологій, продаж товарів та надання онлайн послуг різко підвищився. Інтернет магазини, як підвид сайтів, стали невід'ємною складовою в діяльності людини. Що дозволило зменшити час на відвідування та пошук необхідних товарів в магазині. Приклад популярного магазину, зручної для користування сторінки (див. рисунок 1.2).

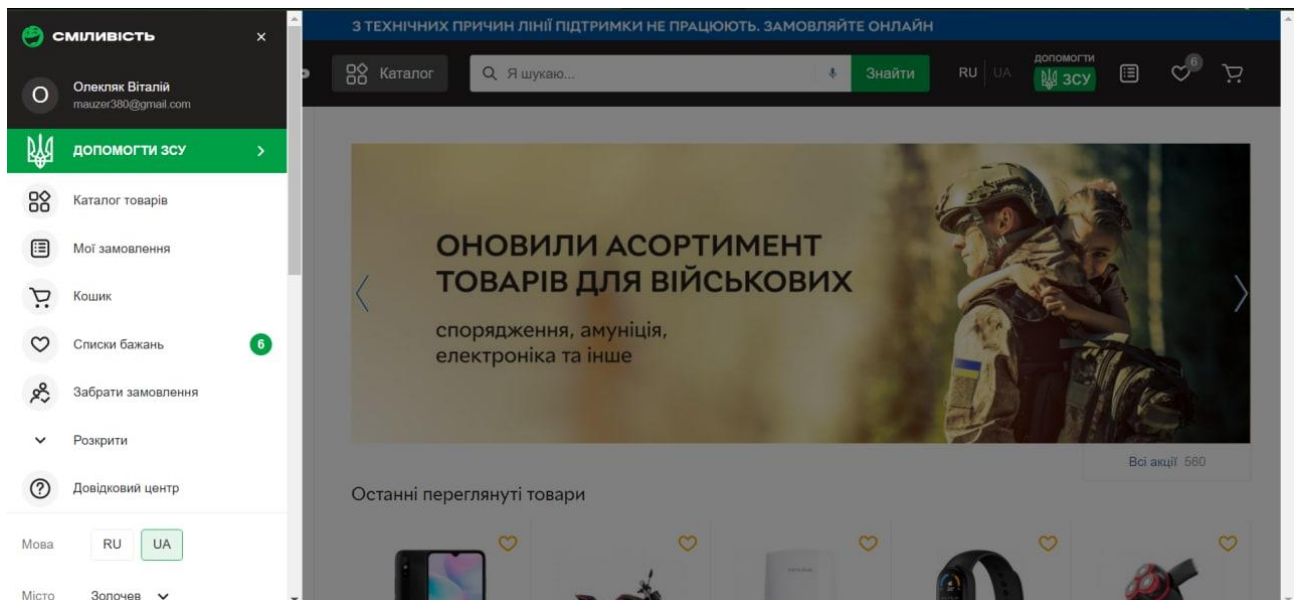


Рисунок 1.2 – Приклад онлайн магазину

А за допомогою пошукового запиту перегляд сайту онлайн магазину дозволить швидко знайти необхідну річ та пришвидшити процес. Замовлення товару, який потрібен можна зробити на сайті та оплатити за нього, також є можливість в онлайн режимі або після доставки його.

1.3 Етапи та структура створення веб-сайтів

При створенні сайту необхідно детально обдумати загальну структуру, від неї залежить практично все: кількість відвідувачів, зацікавленість у перегляді сторінки, зручність при переході на іншу сторінку. Структура сайту та зовнішність вигляд кожної з сторінок є індивідуальними і не схожими на інші, але в Інтернеті можна знайти і такі веб-сторінки, частина функцій та структура яких схожа на інші сторінки в мережі. При перегляді будь-якого сайту можна сказати, що сторінка, яку бачить відвідувач або майбутній клієнт називають головною.

На головній сторінці розміщена вся необхідна інформація, яка міститься на усіх додаткових та проміжних сторінках. Вказана мета з якою працює сайт для чого він призначений і що він містить. Наповнення сайту може бути різним в

залежності від специфікації його тематики та його використання. Найбільшу увагу приділяють головній сторінці, від неї залежить залишиться відвідувач сайту на сторінці чи покине її. Згідно з досліджень, було визначено, користувачі мережі при перегляді веб сторінок не здатні уважно прочитати інформацію з монітору, як з будь-якої іншої книжки. Принцип полягає в тому, що головна сторінка, відповідає вимогам клієнта він залишається і шукає необхідну йому інформацію. Якщо інформації, яку шукає клієнт немає на сторінці, він її залишає і шукає на інших веб-сторінках. Тому головна сторінка робиться спеціально, оскільки, короткий опис відображається у браузері при пошуку інформації.

Для забезпечення швидкого переходу між розділами на веб-сторінці створюють меню. Так званий список гіперпосилань на розділи. Популярним розташуванням меню є у горизонтальному вигляді, шапці сайту. Меню одним із головних елементів сайту. Користувач звертає увагу на меню, завдяки ньому переходить куди йому потрібно. Вимоги до меню є не значними, насамперед кожен розділ необхідно виділити, відокремити від інших, назви повинні бути короткими, помітними та зрозумілими [9].

Сайти, які містять велику кількість інформації можуть містити до десяти сторінок, або переходити за посиланням на інші сторінки. Інформація, яка займає велику кількість місця на сторінках відноситься до третього рівня, а якщо необхідно то й вище [6].

Загалом поділяють структуру сайту на лінійну, дерево подібну, а також довільну. Лінійна структура принцип роботи, якої у переході між сторінками від головної на другу від неї на третю, і так далі. Дерево подібна структура з головної сторінки переходиш на сторінку другого рівня та вище. Довільна структура виглядає дуже хаотично, але саме в цьому полягає принцип реалізації. Перебуваючи на такому сайті з цією структурою, можна з легкістю переходити на будь-яку іншу сторінку. Кожна з представлених структур має свою перевагу і відповідно недоліки.

При розробці сайту першим важливим моментом є реалізація наступних етапів:

- Аналіз структури сайту та побудова семантичного ядра для створення певних алгоритмів, за якими будуються категорії та розділи ресурсу, проводиться оптимізація для систем пошуку.
- Побудова ієрархічної структури, різновиди з покращеними показниками вкладеності, розробка розділів ресурсу та дизайну, налаштування інтеграції для удосконалення якості сайту та кращої взаємодії з базами даних.
- Удосконалення посилань всередині ресурсу, легка і зрозуміла навігація.
- Пошук відповідного контенту, заповнення ним сайту та тестування, ґрунтовний аналіз ресурсу для економії зусиль і часу [19].

Розділів сторінки повинні відповідати вимогам. Тобто, розділ не може містити перевищувати трьох сторінок в одній із категорій. Це дозволить користувачу легко переходити на інші сторінки, зручність та зовнішній вигляд теж відіграють не останню роль. Подивитись архітектуру сайту в конкурентів, який зараз користується попитом дозволяє отримати досвід та підвищити статистику відвідування сайту.

1.4 Платформа для швидкої розробки сайту

Існує величезна кількість платформ, які допомагають полегшувати розробку сайтів користувачеві. Платформи працюють за принципом шаблонів, для їх використання не потрібно досвіду роботи. Однією з перших таких платформ була розроблена компанією «Бітрікс» в 2000 роках. Початком було розробка на замовлення систем керування вмістом. Продукція компанії дозволяє імпортувати користувачів, їх звіти та інше. Але при цьому теж виникають свої складнощі. Іноді вони можуть виникати при оптимізації нової версії продукту. Якщо дані відправляються на віддалений сервер, затримка при обробці може збільшитись. Платформа Bitrix має можливості поєднувати два різні між собою проєкти. Перевагою платформи має потужну систему захисту, спам-атаки, злом,

та інші хакерські дії є мінімальними. Відбувається резервне копіювання даних, відновлення яких виконується автоматично [24].

Найпопулярнішою платформою для створення сайтів та ведення автономних блогів є WordPress. Перша поява відбулась в 2001, саме в той час Мішель Валдрігі розпочав створювати движок, який по сьогоднішній день буде в основі CMS WordPress і зараз його використовують. Оновлення платформи відбувалось практично кожен рік. Можливості, які надає движок платформи досить високо оцінили професіонали. Історія WordPress розповідає як використовують платформу відкритого коду для корисного використання програмного забезпечення. Згодом платформа отримала оптимізацію і можливість використовувати плагіни, які ефективно почали використовувати [7].

Після п'яти оновлень платформи, вона стала гнучкою для користувача. WordPress пристосовується до користувача, функціонал є зручним та простим для роботи з платформою. З'явилась можливість використовувати веб-журнал, перекидати їх у один файл.

WordPress – належить до систем керування вмістом, особливістю якої є програмний вихідний код, повністю безкоштовним та вільний для охочих реалізувати свій сайт. Двигун цієї дивовижної платформи написаний на PHP, беручи за базу даних MySQL. Застосовують в більшості для блогів, онлайн магазинів чи інформаційних сайтів. Розробники вважають, платформа є найефективнішою для цього виду сайтів. Користувачі за час використання платформи, погодились з цією думкою [10].

Платформа здатна зберігати історію змін, що дає переваги користувачам, які часто виправляють і змінюють свою структуру та зовнішній вигляд сайту. Якщо часто змінювати структуру, то легко можна зробити помилку, але історія дозволить її виправити. WordPress автоматизовано зберігає все, що відбувається в процесі роботи користувача та платформи.

Підсумком усього описаного, можна перелічити ряд переваг, даної платформи. WordPress – належить до безкоштовних платформ. Початківець, який бажає розробити сайт, платформа пропонує ряд можливостей, гнучка для

використання, функціонал не потребує навичок та вмінь. Усі процеси є автоматизовані. Встановлення займає не більше п'яти хвилин і не потрібно мати певний рівень знань. Розробники зробили практично все щоб платформа мала простий функціонал. WordPress встановлюється на особистому сервері. Це дає можливість керувати вмістом свого сайту з інших ноутбуків, персональних комп'ютерів, за умови, що вони мають доступ до мережі Інтернет [13].

Список основних переваг платформи представлено:

- наявність величезної бібліотеки якісних платних тем, які володіють більш просунутим функціоналом і гнучкістю настройки;
- підтримка віджетів і соціальних плагінів для поліпшення читабельності і відвідуваності вашого ресурсу;
- надійність і захист системи від злом.

1.5 Застосування ботів в діяльності людини

Бот – програма написана в більшості на мові програмування, є автоматизованою та виконує поставлені дії. Загалом принцип використання ботів полягає, у роботі яка є одноманітною. Використання ботів є невід'ємною складовою у сучасному світі інноваційних технологій. Поведінка та дії, чітко описанні у його коді при розробці [27].

Широке застосування ботів у сайтах, на яких потрібно швидко відповідати клієнтові, надавати послуги, отримувати замовлення чи відправляти зазначені дані тощо. Боти також використовують у стратегічному плані, оскільки розум людини не здатний обраховувати швидко дані, як штучний механізм. Це робить бота однією з потребуючих технологій в сучасному світі. Боти поділяються між собою за призначенням, схильністю до кращої реакції, обрахуванням, залежить від його коду, написаним розробником. Соціальні мережі, групи та онлайн магазини – саме ці середовища є найпопулярнішим у застосуванні ботів.

Боти є незамінною програмою, яка спрощує буденну роботу в діяльності людини. Існують помічники, які запрограмовані голосовими командами. Бот найкраще підходить для сайту продажів товарів та послуг.

1.6 Мова програмування Python, як база для розробки ботів

Python – мова програмування з величезною кількістю можливостей для застосування, дозволяє ефективно та зручно написати код, який читається для інших програм.

Завдяки коду, який буде лаконічним, це дає можливість розробити програму, яка буде ефективнішою і зрозумілішою від свого аналога. Мова програмування Python дозволяє запускати програми незалежно від операційної системи. Стандартна бібліотека мови спільно зв'язана між собою, що дає можливість користувачу використовувати усі інструменти при роботі з операційною системою, браузером, додатками, базами даних, реалізувати процес створення графіки програм [3].

Програми, які були створені за допомогою Python можуть бути простими так і складними, з великою кількістю скриптів. Мова програмування Python є безкоштовною для усіх бажаючих її використовувати. Особливістю є те, що мова не є компільованою. Код, який був написаний на Python спочатку перетворюється на байт-код, а уже згодом інтерпретується мовою програмування [4].

Якщо описати переваги мови Python варто зазначити низький поріг для входу, бібліотеки є зв'язані між собою. Python належить до мов з яким синтаксисом. Читати код є досить просто, оскільки код не містить додаткових елементів. Саме через те що код є без спеціальних елементів, читати його легше. Python є зручним для написання ботів з метою їх реалізації на сайті.

Дану мову програмування було обрано, тому що вона найкраще підходить для проєктування телеграм-бота і досить проста у застосуванні.

1.7 Постановка задачі на розробку

На підставі проведеного аналізу відомих рішень зроблено висновки, що тема роботи є актуальною. Завдяки аналізу предметної області можна виділити основні і допоміжні етапи роботи, також зрозуміти, які елементи повинні брати участь у створенні продукту, тому темою роботи є: дослідити основні принципи створення веб-сайту за допомогою CMS платформи Wordpress 5.8.4, та розробити компонент для замовлення страв, який буде задовольняти вимоги користувачів та замовників, щоб досягти даної мети необхідно виконати наступні задачі:

- провести аналітичний огляд предметної області;
- проаналізувати засоби створення сайтів та телеграм-бота;
- дослідити і проаналізувати роботу CMS платформи WordPress;
- спроектувати структуру сайту;
- розробити веб додаток «Рандеву»;
- розробити телеграм-бота на мові програмування Python;
- загрузити бота на сервер;
- провести тестування розробленого продукту.

1.8 Висновки до першого розділу

В першому розділі було проведено аналіз відомих рішень за темою та виділено основні елементи які допоможуть досягти мети, а саме дослідження основних принципів створення веб-сайту за допомогою CMS платформи Wordpress 5.8.4 з розробленням компонента для замовлення страв, який буде задовольняти вимоги користувачів та замовників.

РОЗДІЛ 2. СТВОРЕННЯ ВЕБ-САЙТУ «РАНДЕВУ» НА CMS ПЛАТФОРМИ WORDPRESS

2.1 Реєстрація домену

Для початку, щоб створити захищений сайт потрібно для початку зареєструвати домен, адже це ім'я ресурсу, що допомагає його ідентифікувати в інтернеті і зареєструвати хостинг, іншими словами орендувати місце на сервері, де знаходитиметься сайт (див. рисунок 2.1). Для створення даного сайту, було вибрано таке доменне ім'я: randevuzl.com.ua [16].

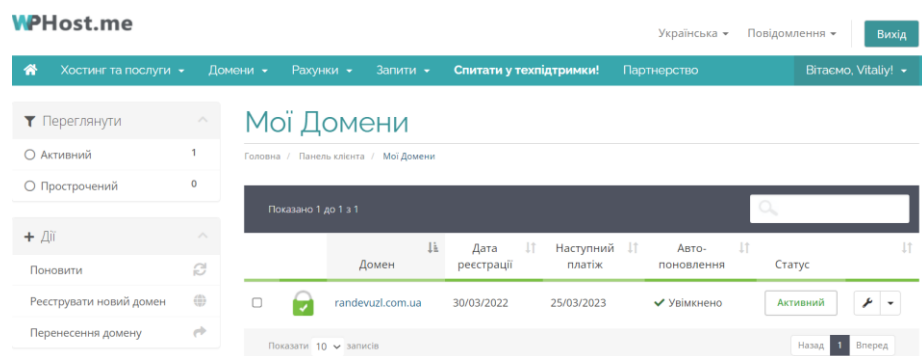


Рисунок 2.1 – Реєстрація домену

Одразу після реєстрації домену було замовлено спеціалізований хостинг на wphost.me [12], який є оптимізований для Wordpress (див. рисунок 2.2), тобто для створення сайту, платформу Wordpress було автоматично встановлено [21].

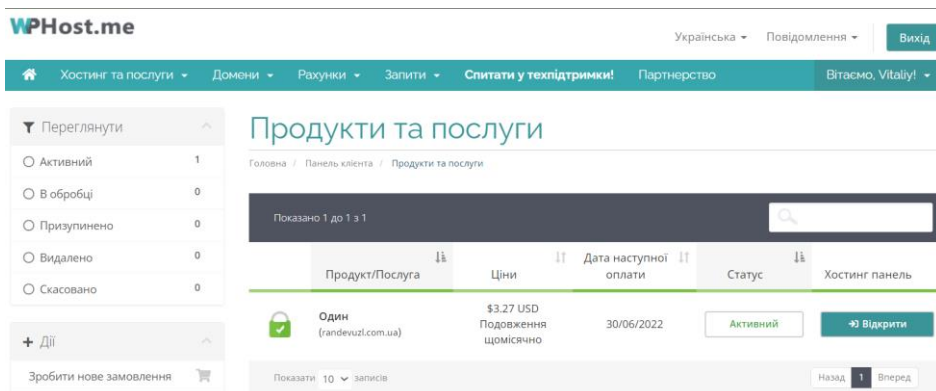


Рисунок 2.2 – Хостинг Wordpress

Після оплати домену і хостингу (див. рисунок 2.3) можна було перейти у свій особистий кабінет.

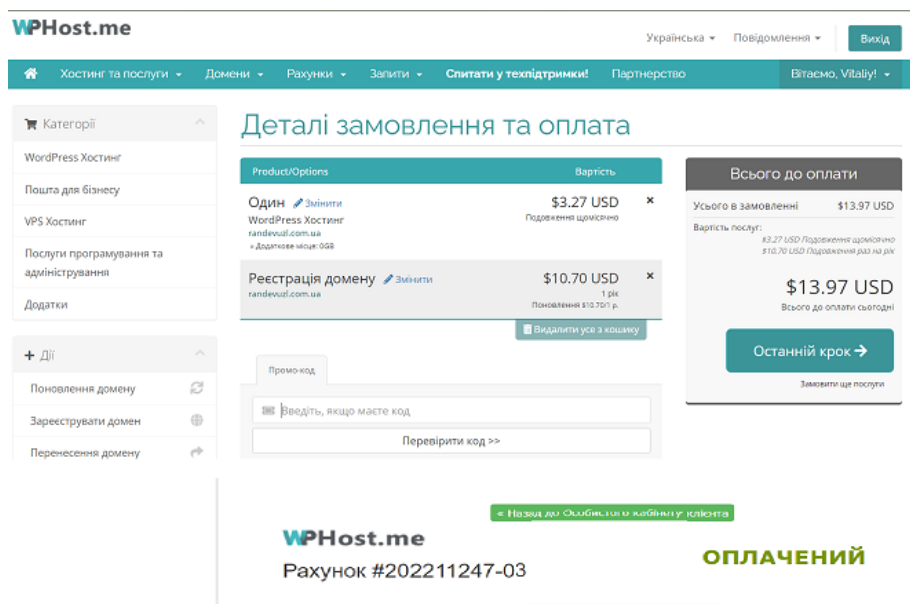


Рисунок 2.3 – Оплата замовлення

У кабінеті можна переглянути замовлені домени, продукти та послуги, а також на головній сторінці (див. рисунок 2.4) побачити активні продукти/послуги чи можливі несплачені рахунки, а ще – замовити нові.

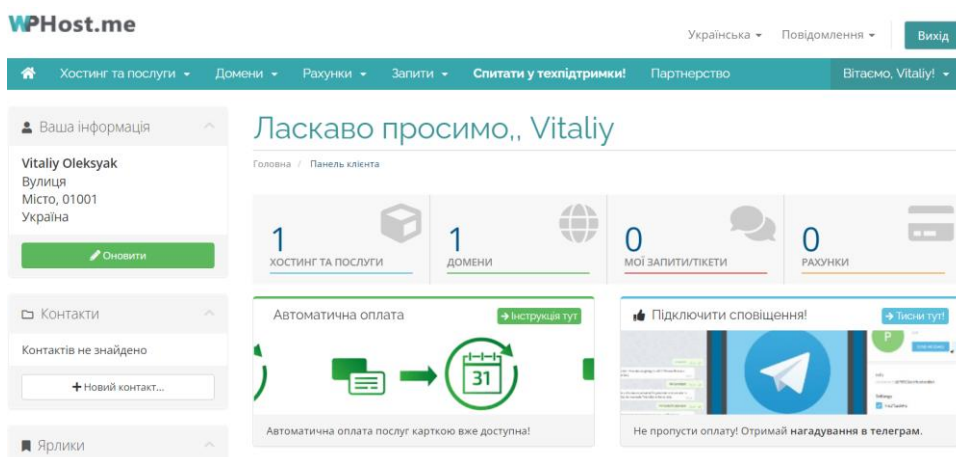


Рисунок 2.4 – Головна сторінка особистого кабінету

Наступним кроком було замовлення SSL сертифіката який необхідний для безпеки сайту і щоб пошукові браузерери могли легко знаходити мій сайт [15].

2.2 Налаштування сайту

Важливим кроком створення сайту є його налаштування. Заходимо у систему WordPress і обираємо налаштування. Для початку, було придумано назву сайту «Randevu ZL», далі було налаштовано мову сайту, формат дати і часу та адресу сайту [14] (див. рисунок 2.5).

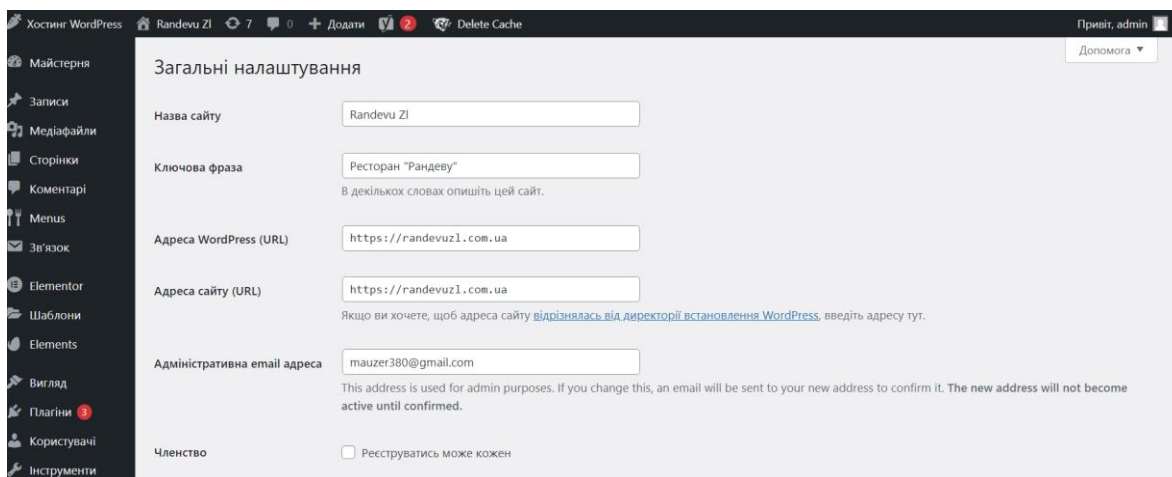


Рисунок 2.5 – Налаштування WordPress

Після того можна переходити до вигляду сторінки, він має бути простим та зручним користувачу. Для цього необхідно перейти в «Вигляд» і обрати пункт «Теми» (див. рисунок 2.6). В кожній версії Wordpress є кілька запропонованих готових тем, можна обрати ті чи завантажити інші.

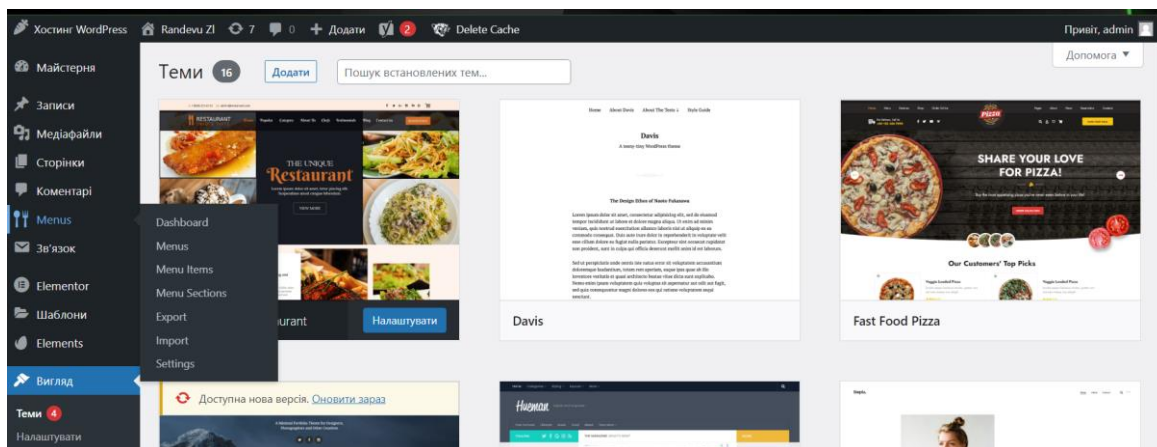


Рисунок 2.6 – Вибір теми

У системі WordPress є допоміжні програми – плагіни, завдяки їм можна редагувати зовнішній вигляд сторінки, текст, додавати меню, анімації та інше. Для зручності встановлено плагіни Elementor, Rating-Widget: Star Review System і Five Star Restaurant Menu (див. рисунок 2.7).

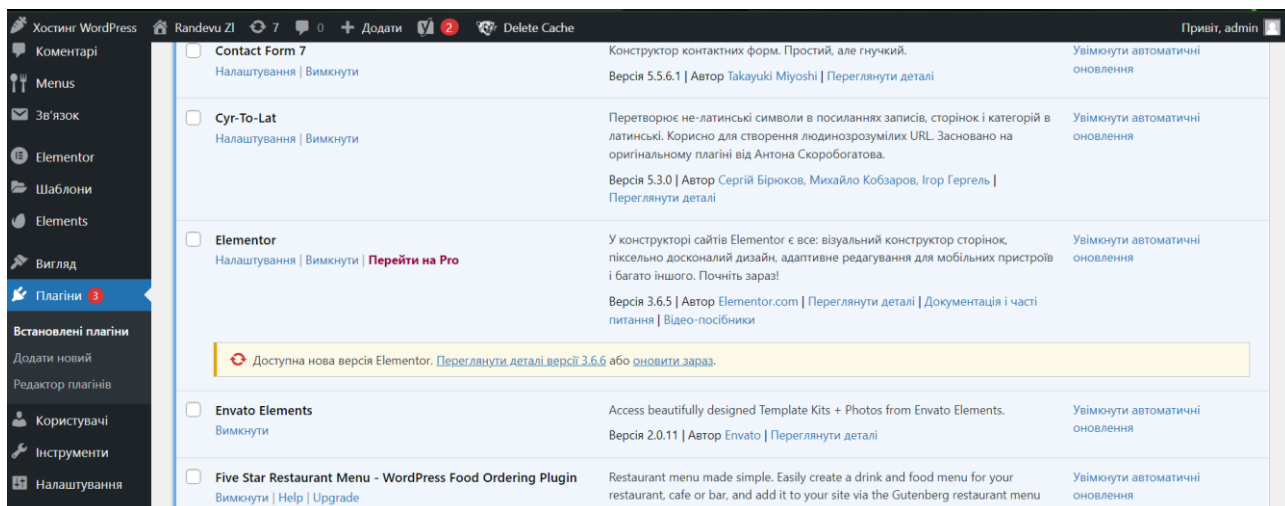


Рисунок 2.8 – Встановлення плагінів

Наступним важливим кроком було створення меню. Для цього потрібно перейти в «Налаштувати» і обрати пункт «Меню» (див. рисунок 2.9). В цьому пункті є готові приклади сторінок, які можна додати в меню, а можна додати власні. Після того, як сторінки були додані, їх можна переміщати в будь-якому порядку.

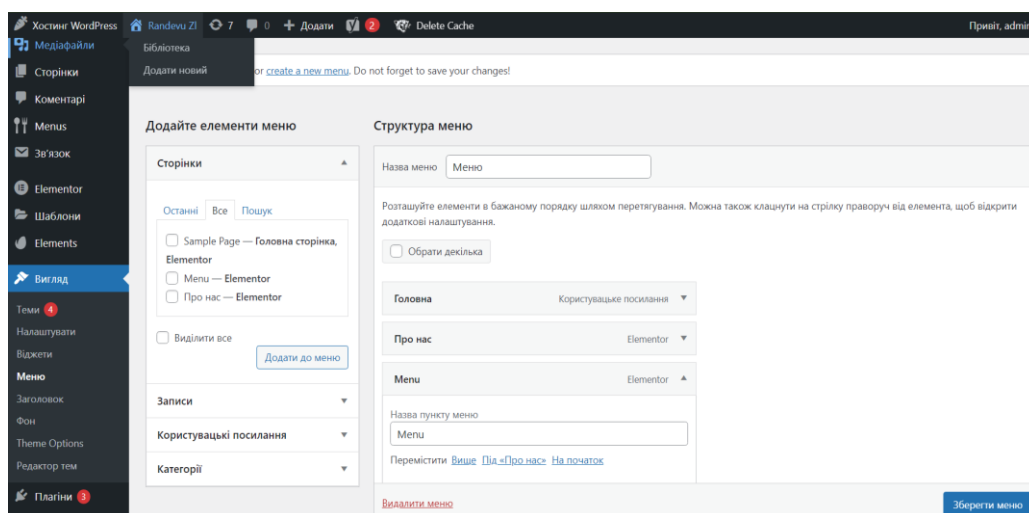


Рисунок 2.10 – Редагування меню

Обравши тему можна переходити до створення сторінок. Щоб створити сторінку потрібно перейти в пункт сторінки, далі обрати створити сторінку і почати оформляти її за допомогою інструментів. Даний сайт створюється для ресторану рандеву тому було створено три основні сторінки: головна, маню і про нас (див. рисунок 2.11).

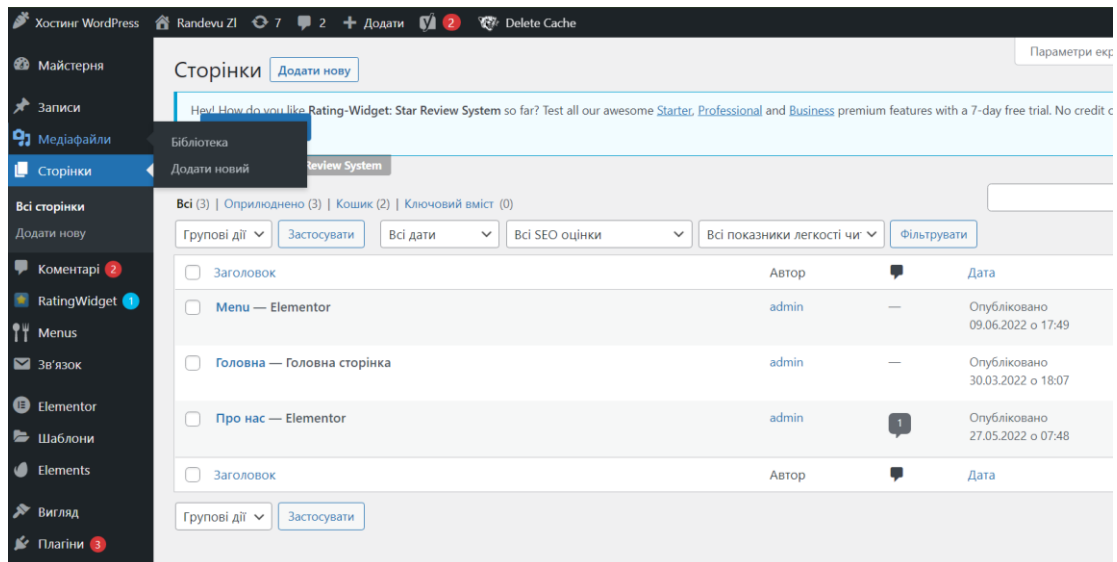


Рисунок 2.11 – Створення сторінок

Основною сторінкою будь-якого сайту є головна сторінка, яка оформлялась за допомогою плагінів, на даній сторінці було вирішено опублікувати посилання на телеграм-бота, який відповість на всі ваші запитання, оформить заказ і доставку (див. рисунок 2.12). Вигляд даної сторінки показано в Додатку Б.

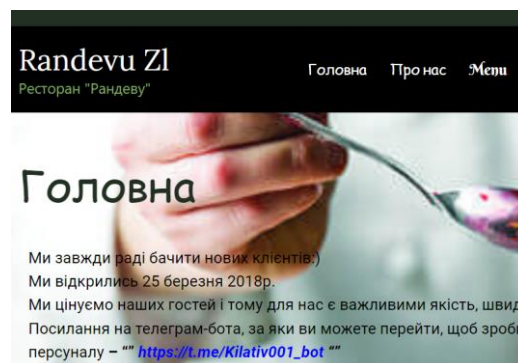


Рисунок 2.12 – Посилання на телеграм-бота

Оформлення сторінки меню. Саме меню було створено за допомогою плагіну – Five Star Restaurant Menu, який значно спростило створення і заповнення самого меню. Меню розділене на розділи, кожен розділ має свої блюда, на кожне блюдо є назва, опис, ціна і фотографія.

В пункті сторінки вибираємо сторінку, меню і вносимо необхідні налаштування для даної сторінки, далі сам вигляд редагуємо за допомогою плагіну Elementor, корегуємо текст і розмір фото, оформивши дану сторінку зберігаємо її і переходимо до наступної (див. рисунок 2.13).

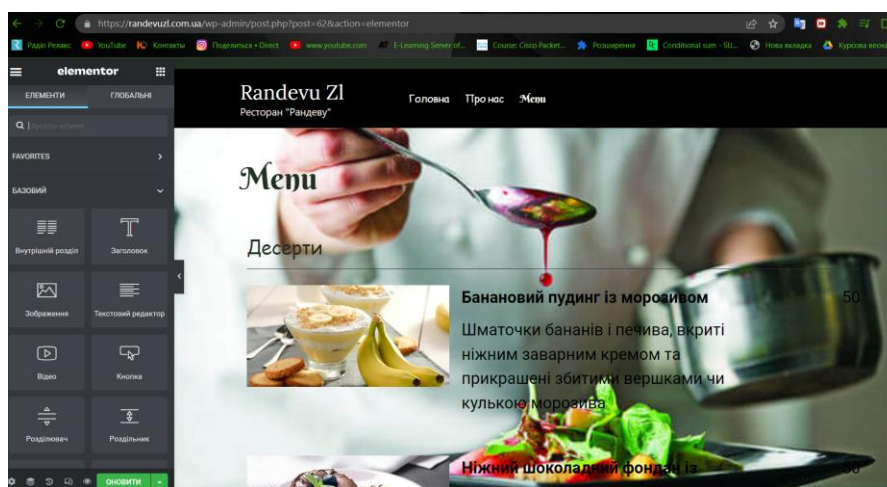


Рисунок 2.13 – Створення сторінок Меню

Подібним способом можна оформити сторінку «Про нас», в якій коротко описано про сам ресторан «Рандеву», його місцезнаходження, коментарі і рейтингова шкала, показано на Додатку Г.

2.3 Розробка телеграм бота

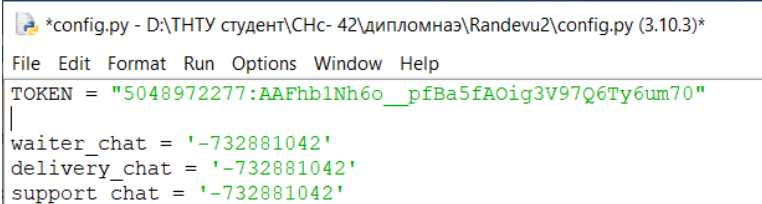
Для замовлення страв було створено телеграм-бот. Для початку, щоб створити телеграм-бота, необхідно в телеграмі запусити BotFather, який запропонує список команд для створення власного бота.

Щоб створити бота потрібно вписати команду /newbot. Далі бот запропонує написати ім'я нового бота, яке будуть бачити користувачі. Після

потрібно придумати username, воно повинне закінчуватись на слово bot. В кінці коли бот буде створений, прийде повідомлення з посиланням на нового бота, також прийде маркер для під'єднання до HTTP API. При створенні бота зразу можна поміняти його аватар та приступити до коду.

2.3.1 Написання коду для телеграм-бота

Для початку потрібно створити файл config.py, куди вписуємо токени бота а саме основний маркер чат-бота і другорядні (див. рисунок 2.14) і другий файл main.py, у якому будуть написані основні команди.



```
*config.py - D:\ТНТУ студент\СНС- 42\дипломназ\Randevu2\config.py (3.10.3)*
File Edit Format Run Options Window Help
TOKEN = "5048972277:AAFhb1Nh6o_pFba5fAOig3V97Q6Ty6um70"
|
waiter_chat = '-732881042'
delivery_chat = '-732881042'
support_chat = '-732881042'
```

Рисунок 2.14 – Token бота

Далі потрібно імпортувати файл з токеном, базу даних та бібліотеки. Було створено наступні команди /cancel, яка вимикатиме бот, /start яка запускатиме бот і /support_answer за допомогою якої можна буде відповідати на запитання з підтримки. В додаток було створену групу Randevu ZL в телеграмі, запросивши в неї IDbot він відобразить id даної групи, яке потрібно вставити в файл config.py, щоб заклази, бронювання і підтримка відображалась в цій групі.

2.3.2 Тестування

Запустивши бота і ввівши команду /start, він відповість на неї привітним повідомленням, а саме: «Давно не бачились, давайте я нагадаю про себе») і запитає що ви бажаєте зробити, також повідомить вас що він інколи може втомитись і не відповідати на команди, тому потрібно ввести команду /cancel,

яка дозволить перезапустити бота, і він знову зможе відповідати на команди (див. рисунок 2.15).

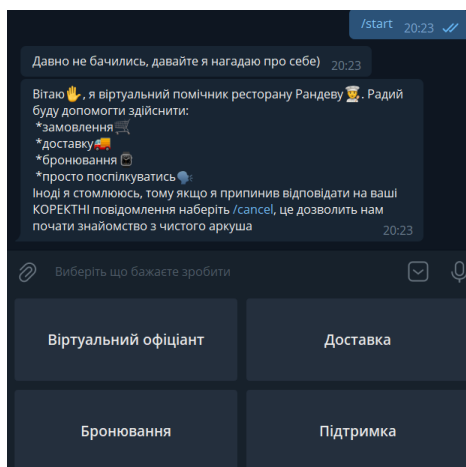


Рисунок 2.14 – Результат команди /start

Якщо обрати віртуальний офіціант бот запитатиме у вас за яким столиком бажаєте сидіти, далі що ви хочете замовити: гарячі страви, холодні страви, напої чи десерт, обравши щось з меню він запитатиме чи не бажаєте ще чогось, обравши ні він відобразить ціну замовлення, якщо так то потрібно обрати іншу страву з меню. Після замовлення в групі Randevo ZL відобразиться даний замовлення (див. рисунок 2.15).

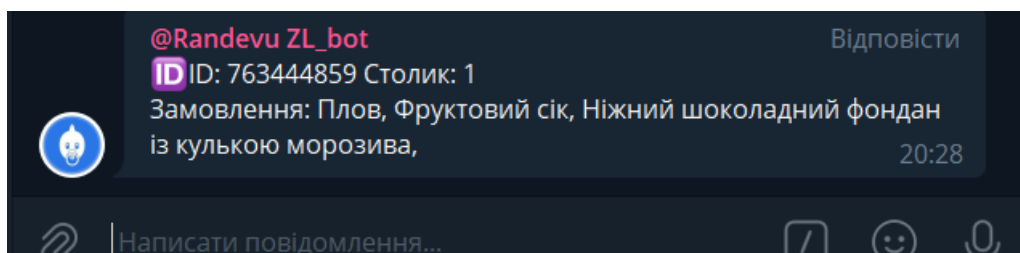


Рисунок 2.15 – Результат дії віртуальний офіціант

Наступна дія бронювання, спочатку бот запитатиме вас дату бронювання, якщо на цю дату вже заброньований столик він попросить вибрати інший, далі потрібно визначити годину з 10 по 23, вказати скільки осіб і столик, в кінці бронювання він уточнить його (див. рисунок 2.16), також дані зберігаються в автоматично створеній БД.

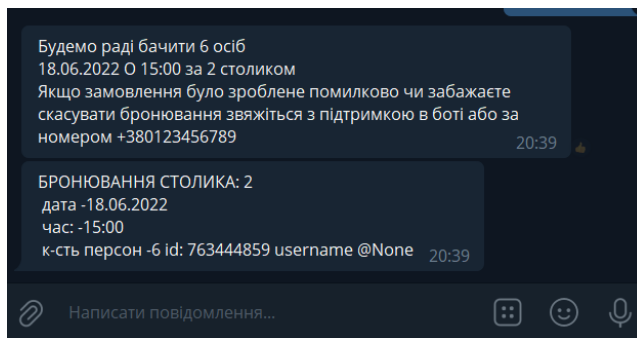


Рисунок 2.16 – Результат дії бронювання

Дія доставка працює наступним чином, бот запитує у вас телефон, адрес і що бажаєте замовити з меню потім в групі Randevu ZL відобразиться доставка (див. рисунок 2.17).

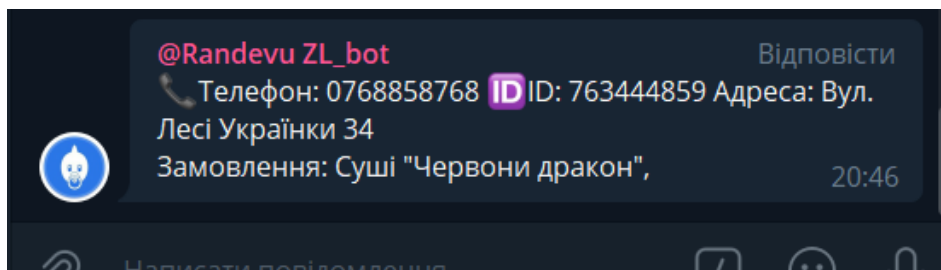


Рисунок 2.17 – Результат дії доставка

І остання дія це підтримка, за допомогою якої можна задати запитання обслуговуванню на пряму. Обравши дію підтримка бот запропонує написати своє питання, після його відправлення попросить почекати на відповідь. Давши відповідь через команду `/support_answer`, бот відобразить її (див. рисунок 2.18).

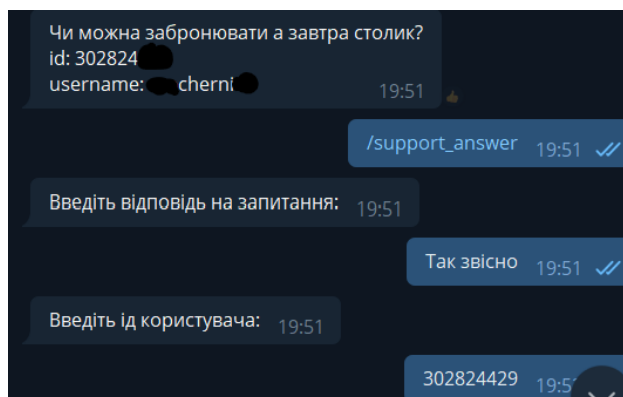


Рисунок 2.18 – Результат дії підтримка

Блок-схема роботи телеграм бота вказана в Додатку В, код написаний на Python відображено в Додатку А. Код оформлення меню можна побачити в Додатку Д.

2.3.3 Загрузка бота на сервер

Загрузка бота позволит сэкономить ресурсы компьютера. Даний бот було загружено на сайт `pythonanywhere`, спочатку загружаємо всі необхідні файли на сервер далі за допомогою команди `/python main.py` запускаємо бот (див. рисунок 2.19).

```
16:27 ~/RandevuzL $ python main.py  
2022-06-15 16:28:14,174 - apscheduler.scheduler - INFO - Scheduler started
```

Рисунок 2.19 – Виконання команди `/python main.py`

2.4 Висновки до другого розділу

В другому розділі було описано створення сайту його налаштування і оформлення зовнішнього вигляду, розроблено телеграм-бота, проведено його тестування і завантажено на сервер.

РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

3.1 Долікарська допомога при харчових отруєннях

Практично кожна людина, хоч раз у своєму житті, відчувала на собі прояви харчового отруєння. Особливо часто таке негативне явище виникає в дітей. Тому важливо знати, що робити й чого боятися в разі неприємності. І, звичайно, як уникнути отруєння в майбутньому.

Харчове отруєння виникає, коли людина вживає в їжу будь-які продукти, обсіменені хвороботворними бактеріями. Список патогенних мікроорганізмів дуже широкий, і багато хто з них при попаданні в травну систему може призводити до дуже тяжких наслідків.

Проблема ускладнюється ще тим, що патогенні мікроорганізми, перебуваючи в продуктах харчування, продукують токсини. Ці речовини накопичуються в харчах та викликають симптоми отруєння, навіть коли їжа досить термічно оброблена й живих мікробних тіл у ній вже немає.

Хоча у багатьох продуктах є потенційно шкідливі організми, зазвичай вони гинуть під час приготування страви.

Але якщо ви не дотримуєтеся правил гігієни, неправильно зберігаєте продукти, навіть приготовлені страви можуть призвести до хвороби.

Продуктів які становлять ризик харчового отруєння є багато. Найбільш небезпечні ті, які повинні зберігатися в прохолодному місці, у холодильниках і морозильниках. Спекотної пори року, такі умови забезпечити вдається не завжди, особливо, коли компанія або сім'я виходить на пікнік на природу.

Небезпеку в плані харчового отруєння становлять: сире або недоварене м'ясо або птиця; непастеризовані молочні продукти; сирі молюски, риба, морепродукти; немиті фрукти та овочі.

Є десять основних симптом харчового отруєння:

- біль у животі та спазми;
- діарея;

- головний біль;
- рвота;
- загальне відчуття нездужання;
- лихоманка;
- озноб (морозить);
- слабкість і втома;
- нудота;
- біль у м'язах.

Щоб запобігти харчовим отруєнням необхідно:

- ретельно мийте фрукти й овочі;
- ретельно мийте стільниці, обробні дошки, ножі й посуд, перш ніж починати застосовувати їх для приготування їжі;
- часто мийте руки й посуд;
- не кладіть сире і приготоване м'ясо на одну тарілку;
- якщо ножі використовувалися для оброблення сирової курки, не використовуйте їх для інших продуктів, які не будуть піддаватися термічній обробці;
- готуйте м'ясо ретельно, у разі необхідності використовуйте термометр для мяса;
- не використовуйте розфасовані продукти, термін придатності яких минув;
- не їсти м'які сири при вагітності або в разі слабого імунітету [26].

3.2 Протипожежні вимоги до виробничого освітлення

У системах електрозабезпечення можуть виникати режими, що характеризуються електричними, тепловими, механічними навантаженнями, які значно перевищують навантаження нормального режиму роботи. При значному перенавантаженні електричних приладів струмами або при короткому замиканні виділяється велика кількість теплової енергії, яка загрожує збереженню

установки та можливості виникнення пожежі. Основним видом аварій у системах електрозабезпечення є коротке замикання, через це апарат від'єднання повинен володіти відповідною можливістю від'єднувати. У вигляді таких апаратів можна застосовувати плавкі запобіжники НН (низької напруги), автоматичні вимикачі, які керуються при аварійному вимкненні, вмонтованому у привід роз'єднувачами або релейним захистом. Залежно від застосованої апаратури захист може реагувати на зміну визначених електричних або неелектричних величин: струму, напруги, потужності, опору, кута зсуву фаз, частоти, температури, випромінювання, рівня, потоку газу або рідини. Апарати, які застосовують у системах електричного захисту в електромережах умовно поділяють на дві групи: захист плавкими запобіжниками; захист автоматичними вимикачами.

Займання в приміщенні може бути викликано необережним поведінням з електричними приладами або з вогнем. Для того, щоб не сталося біди, треба дотримуватися вимог пожежної безпеки до приміщень.

Запобіжники призначені для захисту електричних мереж від перенавантажень і короткого замикання. Найбільше розповсюдження отримали плавкі запобіжники, вони дешеві та прості за будовою. Плавкий запобіжник складається з двох основних пластин: корпусу (патрона) із електроізоляційного матеріалу та плавкої вставки (мідь, алюміній, свинець). Кінці плавкої вставки з'єднані з клемами, за допомогою яких запобіжник вмикається в лінію послідовно зі споживачами, які захищають ділянку кола. Плавку вставку вибирають з таким розрахунком, щоб вона розплавилась раніше, ніж температура проводу мережі досягне небезпечного рівня чи перенавантажений споживач вийде з ладу.

Елементарні правила пожежної безпеки в приміщенні:

- не викидати у відро для сміття гарячі сірники, недопалки або тліючу папір;
- уникати паління в підвалах, сараях, коморах або в ліжку;
- не користуватися вогнем в приміщеннях, де присутні матеріали,

здатні швидко спалахнути;

- не допускати дітей до розеток і вимикача світла [19].

3.3 Висновки до третього розділу

Проаналізовано аналіз двох питань, а саме долікарська допомога при харчових отруєннях і протипожежні вимоги до виробничого освітлення зроблено наступні висновки.

На сьогоднішній день є багато випадків отруєння а саме в ресторані кафе у дома, тому необхідно мати базові знання щоб надати першу медичну допомогу як собі так і близьким, бо його наслідки можуть бути фатальними. При отруєнні необхідно звертатися зразу до лікарів, оскільки саме вони зможуть надати кваліфіковану допомогу. Перед приїздом швидкої, особі яка отримала харчове отруєння, необхідно давати багато води.

На рахунок пожеж які виникають з причин пошкодження електромережі, необхідно знати і врахувати ряд дій які зможуть врятувати життя і попередити виникнення нових пожеж. Для попередження пожеж вогнегасники розставити у визначених місцях, а також проведено протипожежну систему, інструктаж з персоналом на тему протипожежної безпеки, оскільки неймовірна кількість пожеж виникає саме з халатності людей.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи, отримано повнофункціональний веб-сайт, готовий до використання користувачами у режимі онлайн. Сайт «Рандеву» розроблений на CMS платформі Wordpress 5.8.4, також розроблений телеграм-бот, за допомогою якого можна зробити заказ, забронювати столик, доставку і задати питання на пряму персону.

В першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр»:

- описано етапи та структури створення сайтів;
- розглянуто основні питання створення телеграм-бота.

В другому розділі кваліфікаційної роботи:

- створено сайт «Рандеву»;
- досліджено і проаналізовано роботу CMS платформи WordPress 5.8.4;
- розроблено аналіз засобів створення сайтів та телеграм-бота;
- розроблено телеграм-бота на мові програмування Python;
- завантажено бота на сервер.

У розділі «Безпека життєдіяльності, основи хорони праці» висвітлено питання долікарської допомога при харчових отруєннях та протипожежних вимоги до виробничого освітлення.

ПЕРЕЛІК ДЖЕРЕЛ

1. Дакетт Д. А. HTML и CSS. Разработка и дизайн веб-сайтов / Д. А. Дакетт. – Эксмо, 2020. – 480 с.
2. HTML мова розмітки гіпертексту [Електронний ресурс]: Режим доступу: <http://www.znannya.org/?view=html>
3. PHP [Електронний ресурс]: Режим доступу: <https://uk.wikipedia.org/wiki/PHP>
4. DB Browser for SQLite. (DB4S) – [Електронний ресурс]: Режим доступу до ресурсу: <https://sqlitebrowser.org/>
5. Дронов В. А. PHP, MySQL и DreamweaverMX 2004. «Разработка интерактивных Web-сайтов.» / В. А. Дронов. – СПб:Петербург, 2005. – 448 с.
6. Wphost.me. Хостинг Wordpress. [Електронний ресурс]: 2020. Режим доступу: <https://wphost.me>
7. Ашманов И.С., Продвижение сайта в поисковых системах / И.С. Ашманов, А.А. Иванов. – М.: Вильямс, 2007. – 304 с.;
8. Гарретта Д. Р. Веб-дизайн. Элементы досвіду взаємодії / Д. Р. Гарретта. – Символ-Плюс, 2008. – 192 с.
9. Нильсена Я. О. Веб-дизайн / Я. О. Нильсена. – Символ-Плюс, 2014. – 512 с.
10. Вікіпедія. WordPress. [Електронний ресурс]: 2020. Режим доступу: <https://uk.wikipedia.org/wiki/WordPress>
11. Вікіпедія. Вебсайт. [Електронний ресурс]: 2020. Режим доступу: <https://uk.wikipedia.org/wiki/Вебсайт>
12. Вікіпедія. Доменне ім'я. [Електронний ресурс]: 2020. Режим доступу: https://uk.wikipedia.org/wiki/Доменне_ім'я
13. Вікіпедія. Хостинг. [Електронний ресурс]: 2019. Режим доступу: <https://uk.wikipedia.org/wiki/Хостинг>
14. Горнаков С. Г. Осваиваем популярные системы управления сайтом (CMS) / С. Г. Горнаков. – М.: ДМК Пресс, 2009. – 336 с.

15. Джон Дакетт. HTML и CSS. Разработка и дизайн веб-сайтов. / Дакетт Джон: Эксмо, 2016. – 481 с.
16. Дунаев В. Основы WEB-дизайна. Самоучитель / В. Дунаев. – Москва: БХВ Петербург, 2014. – 215 с.
17. Купер К. Л. Как создать сайт. Комикс-путеводитель по HTML, CSS и WordPress / К. Л. Купер. – Манн, Иванов и Фербер, 2019. – 264 с.
18. Кудрец Д. Основы CSS. Издательские решения / Д. Кудрец. – Видавничі рішення, 2019. – 32 с.
19. Кудрец Д. Практикум по HTML. Издательские решения / Д. Кудрец. – Видавничі рішення, 2018. – 60 с.
20. Основні етапи створення сайту [Електронний ресурс]: Режим доступу: <https://pbb.lviv.ua/statti-i-novyny/statti-shchodo-stvorennia-saitu/osnovni-etapy-stvorennia-saitu>
21. Основы CSS и HTML. Блочная верстка [Електронний ресурс]: Инструкция по созданию сайта. – Режим доступу: <http://www.web-lesson.ru/site-creation/html/38-osnovy-css-i-html-blochnaya-verstka.html>
22. Офіційний сайт WordPress [Електронний ресурс]: Режим доступу : <https://ru.wordpress.org>
23. Севостьянов И. О. Поисковая оптимизация. Практическое руководство по продвижению сайта в Интернете / И. О. Севостьянов. – Питер: СПб, 2010. – 240 с.
24. Протипожежні вимоги [Електронний ресурс]: Режим доступу: <https://oppb.com.ua/docs/vimogi-pozhezhnoyi-bezpeki-do-utrimannya-evakuaciynih-shlyahiv-i-vihodiv>
25. Сайт по веб-програмуванню “Way to start”. [Електронний ресурс]: Режим доступу: <https://waytostart.ru/blog/telegram-bot>;
26. Сергеев А. Н. Создание сайтов на основе WordPress / А. Н. Сергеев. – ЛАНЬ, 2015. – 128 с.
27. Створення телеграм-бота [Електронний ресурс]: <https://netpeak.net/ua>

28. Типи сайтів [Електронний ресурс]: Режим доступу:
<https://internetdevels.ua/blog/most-common-types-of-websites>
29. Фрейн Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств издание И.О. Севостьянов. – СПб.: Питер, 2017. – 272 с.
30. Харчове отруєння [Електронний ресурс]: Режим доступу:
[//zdrav.ck.gov.ua/uk/content/harchove-otrueynnya-persha-dopomoga-profilaktyk](http://zdrav.ck.gov.ua/uk/content/harchove-otrueynnya-persha-dopomoga-profilaktyk)

ДОДАТКИ

Код телеграм-бота

```

import logging
import config
import sqlite3
import datetime
import random
from datetime import datetime
from telegram import ReplyKeyboardMarkup, ReplyKeyboardRemove,
Update
from telegram.ext import (
    Updater,
    CommandHandler,
    MessageHandler,
    Filters,
    ConversationHandler,
    CallbackContext,
)

# Enable logging
logging.basicConfig(
    format='%(asctime)s - %(name)s - %(levelname)s - %(message)s',
    level=logging.INFO
)

logger = logging.getLogger(__name__)

CHECK, TYPE, MENU, ORDER, DECISION, TABLENUMBER, ADDRESSA,
PHONENUMBER, SUPPORT, SUPPORT_ANSWER, ANSWER, DATE_BOOK,
TIME_BOOK, CUSTOMER_COUNT, TABLE_BOOK = range(
    15)

def start(update: Update, context: CallbackContext) -> int:
    info = (update.message.chat)
    conn = sqlite3.connect('Randevu.db')
    cur = conn.cursor()
    cur.execute("""CREATE TABLE IF NOT EXISTS Users(
        ID INT,
        date_activate TEXT,
        username TEXT,
        firstname TEXT);
    """)
    cur.execute('''Select * from Users where ID=?''',
    (info["id"],))
    rows = cur.fetchall()
    if not rows:
        update.message.reply_text(
            'Бачу поки ми не знайомі, давайте це виправимо')
    with conn:
        cur = conn.cursor()

```

```

        user_id = update.message.chat['id']
        nick = update.message.chat['username']
        first_name = update.message.chat['first_name']
        date = datetime.now()
        cur.execute("""INSERT INTO Users(ID, date_activate,
username, firstname)

        VALUES(?, ?, ?, ?);""", (user_id, str(date), nick,
first_name))
    else:
        update.message.reply_text(
            'Давно не бачились, давайте я нагадаю про себе')
    print(info['id'])
    global my_order
    my_order = []
    reply_keyboard = [['Віртуальний офіціант', 'Доставка'],
                      ['Бронювання', 'Підтримка']]
    update.message.reply_text(
        'Вітаю👋, я віртуальний помічник ресторану Рандеву👨🍳.
Радий буду допомогти здійснити:\n *замовлення🛒 \n *доставку🚚 \n
*бронювання🕒\n *просто поспілкуватись🗣️\nІноді я стомлююсь, тому
якщо я припинив відповідати на ваші КОРЕКТНІ повідомлення наберіть
/cancel, це дозволить нам почати знайомство з чистого аркуша ',
        reply_markup=ReplyKeyboardMarkup(
            reply_keyboard, one_time_keyboard=True,
input_field_placeholder='Виберіть що бажаєте зробити'
        ),
    )
    return CHECK
def check(update: Update, context: CallbackContext) -> int:
    # print(update.message.text)
    choice = update.message.text
    global chat_id
    if choice == 'Віртуальний офіціант':
        chat_id = config.waiter_chat
        id = update.message.chat_id
        my_order.append(id)
        id = update.message.chat_id
        my_order.append(id)
        reply_keyboard = [['1', '2'],
                          ['3', '4']]
        update.message.reply_text(
            'Оберіть номер столика🎯',
            reply_markup=ReplyKeyboardMarkup(
                reply_keyboard, one_time_keyboard=True,
input_field_placeholder='Номер столика:'),)
        return TYPE

    elif choice == 'Доставка':
        chat_id = config.delivery_chat
        update.message.reply_text(

```

```

        '📞 Напишіть свій номер телефону (у форматі
0971234567):')
        return PHONENUMBER

    elif choice == 'Бронювання':
        update.message.reply_text(

            'Я готовий забронювати для вас місце в нашому прекрасному
закладі.\n Для початку введіть на яку дату забронювати 📅
(Наприклад 21.12.2020)')
            return DATE_BOOK

    elif choice == 'Підтримка':
        global user_info
        user_info = update.message.chat
        update.message.reply_text('Напишіть своє питання.')
        return SUPPORT
    else:
        update.message.reply_text('🔊 Зробіть свій вибір із
запропонованого')

def date_book(update: Update, context: CallbackContext) -> int:
    global data
    data = {}
    current_datetime = datetime.now()
    text = update.message.text.split('\n')
    data["date"] = text
    book_date = data["date"][0].split(".")
    # print(book_date) #дата бронювання
    if (int(book_date[1]) == 2 and int(book_date[0]) > 29) or
(int(book_date[1]) % 2 == 0 and int(book_date[0]) > 30):
        update.message.reply_text(
            '🔊 Введена дата не є дійсною , будь-ласка введіть
коректну дату')
        return CHECK
    elif (int(book_date[1]) == 2 and int(book_date[0]) > 29) or
(int(book_date[1]) % 2 == 0 and int(book_date[0]) > 30):
        update.message.reply_text(
            '🔊 Введена дата не є дійсною , будь-ласка введіть
коректну дату')
        return CHECK
    elif int(book_date[2]) > current_datetime.year or
(int(book_date[2]) == current_datetime.year and int(book_date[1])
> current_datetime.month) or (int(book_date[2]) ==
current_datetime.year and int(book_date[1]) ==
current_datetime.month and int(book_date[0]) >
current_datetime.day):
        # print(str(current_datetime.year) +
str(current_datetime.month)+ str(current_datetime.day)) #поточна
дата
        update.message.reply_text(

```

```

        '🕒 В якій годині бажаєте відвідати нас? (формат 20:22
Зауважте бронювання доступне з 10 до 22 )')
        return TIME_BOOK
    else:
        update.message.reply_text(
            '📞 Введена дата не є дійсною , будь-ласка введіть
коректну дату натисніть /start')
        return ConversationHandler.END

def time_book(update: Update, context: CallbackContext) -> int:
    text =
update.message.text.split('\n'
    data["time"] = text
    book_time = data["time"][0].split(":")

    # print(book_time)
    if int(book_time[0]) > 9 and int(book_time[0]) < 23:
        update.message.reply_text('👤 Скільки вас буде? (до 6
осіб)')
        return CUSTOMER_COUNT
    else:
        update.message.reply_text('Вибачте але бронювання доступне
з 10-22')
        update.message.reply_text(
            '🕒 В якій годині бажаєте відвідати нас? (формат 20:22
Зауважте бронювання доступне з 10 до 22 )')
        return TIME_BOOK

def customer_count(update: Update, context: CallbackContext) ->
int:
    # print(update.message.text)
    if int(update.message.text) > 6:
        update.message.reply_text(
            "👤👤 Більше 6 осіб вважається банкетом! \nДля
замовлення банкету дистанційно подзвоніть за номером
+380123456789📞 \nабо зв'яжіться з підтримкою за допомогою даного
бота)")
        return ConversationHandler.END
    else:
        text = update.message.text.split('\n')
        data["count"] = text
        update.message.reply_text(str(data))
        conn = sqlite3.connect('Randevu.db')
        cur = conn.cursor()
        cur.execute('PRAGMA foreign_keys = ON;')
        conn.commit()
        # print(str(data["date"][0]))
        cur.execute(''Select * from Reserve where Число=?'',
(data["date"]))
        result = cur.fetchall()

```

```

# print(result)
i = 0
global table_string
table_string = '1234'
while i != len(result):
    # print(result[i][1])
    if str(result[i][1]) in table_string:
        table_string =
table_string.replace(str(result[i][1]), '')
        i = i+1
    else:
        i = i+1

if table_string != '':
    reply_keyboard = [table_string]
    update.message.reply_text('На вибрану дату вільні
наступні столики, виберіть який вам підходить:
'', reply_markup=ReplyKeyboardMarkup(reply_keyboard,
one_time_keyboard=True))
    return TABLE_BOOK else:

update.message.reply_text(
    '''⊖ На вибрану дату немає вільних столиків''')
update.message.reply_text(
    'Будь ласка оберіть іншу дату(Наприклад 21.12.2020) ,
або відмініть замовлення командою /cancel ')
return DATE_BOOK

def table_book(update: Update, context: CallbackContext) -> int:
    if str(update.message.text) in table_string:
        data["table_number"] = update.message.text
        conn = sqlite3.connect('Randevu.db')
        cur = conn.cursor()
        idu = random.randrange(10, 1000)
        cur.execute("""CREATE TABLE IF NOT EXISTS Reserve(
ід INT PRIMARY KEY,
Номер INT,
Число TEXT,
Час TEXT,
Кількість INT);
""")
        cur.execute("""INSERT INTO Reserve(ід, Номер, Число, Час,
Кількість)
VALUES(?, ?, ?, ?, ?);""", (idu, data['table_number'],
str(data['date'][0]), str(data['time'][0]),
int(data['count'][0])))
        conn.commit()
        update.message.reply_text("Будемо раді бачити " +
str(data['count'][0]) + ' осіб\n' + str(data["date"][0]) + ' о ' +
str(data['time'][0]) + ' за ' + str(
data['table_number'][0]) + ' столиком\n' + 'Якщо
замовлення було зроблене помилково чи забажаєте скасувати

```

```

бронювання зв'яжіться з підтримкою в боті або за номером
+380123456789'))))
    # print(data)
    ans = 'БРОНЮВАННЯ СТОЛИКА: ' + str(data['table_number']) +
'\n дата -' + str(data["date"][0]) + '\n час: -' + str(
    data["time"][0]) + '\n к-сть персон -' +
str(data["count"][0]) + ' id: ' + str(update.message.chat_id) + '
username @' + str(update.message.chat.username)
    chat_id = config.support_chat
    context.bot.send_message(chat_id, ans)
    return ConversationHandler.END
else:
    update.message.reply_text('🔊 Потрібно вибрати щось з
запропонованого')
    reply_keyboard = [table_string]
    update.message.reply_text(''''На вибрану дату вільні
наступні столики, виберіть який вам підходить:
''', reply_markup=ReplyKeyboardMarkup(reply_keyboard,
one_time_keyboard=True))
    return TABLE_BOOK
def support(update: Update, context: CallbackContext) -> int:
    chat_id = config.support_chat
    problem_message = update.message.text
    # print(update.message.chat['id'])
    # print(update.message.chat['username'])
    context.bot.send_message(chat_id, str(problem_message) +
'\nid: ' + str(
        update.message.chat['id']) + '\nusername: ' +
str(update.message.chat['username']))
    update.message.reply_text('🔊 Ваше запитання подане, очікуйте
відповіді.')
    return ConversationHandler.END

def getphonenumber(update: Update, context: CallbackContext) ->
int:
    phone = update.message.text
    id = update.message.chat_id
    print(id)
    my_order.append(phone)
    my_order.append(id)
    update.message.reply_text('🏠 Напишіть свою адресу')
    return ADDRESSA

def getadressa(update: Update, context: CallbackContext) -> int:
    addr = update.message.text
    my_order.append(addr)
    reply_keyboard = [['🍲 Гарячі страви', '🥗 Холодні страви'],
['🍷 Напої', '🍰 Десерти']]
    update.message.reply_text('Справді? А я гуляв там вчора 😊\n Що
бажаєте замовити?', reply_markup=ReplyKeyboardMarkup(

```

```

        reply_keyboard, one_time_keyboard=True,
input_field_placeholder=' Що бажаєте замовити?'
    ),
)
return MENU

# menu

def eat_type(update: Update, context: CallbackContext) -> int:
    my_order.append(update.message.text)
    reply_keyboard = [['🍲 Гарячі страви', '🍧 Холодні страви'],
                      ['🍷 Напої', '🍰 Десерти']]
    update.message.reply_text('Я персональний офіціант👤, що
бажаєте замовити?', reply_markup=ReplyKeyboardMarkup(
        reply_keyboard, one_time_keyboard=True,
input_field_placeholder=' Що бажаєте замовити?'
    ),
)
return MENU

def menu(update: Update, context: CallbackContext) -> int:
choice = update.message.text
print(len(choice))
choice = choice[2:]+choice[:0]
print(len(choice))
conn = sqlite3.connect('Randevu.db')
cur = conn.cursor()
cur.execute('PRAGMA foreign_keys = ON;')
conn.commit()
cur.execute('Select * from меню where Тип=?', (choice,))
result = cur.fetchall()
choice=choice.replace(' ', '')
print(choice)
print(result)
if result != []:
    for row in result:
        update.message.reply_text(str(
            row[1]) + '\n🔍Інгредієнти: ' + str(row[3]) +
'\n💰Ціна: ' + str(row[4]) + ' грн.')
        reply_keyboard = [[str(row[1])] for row in result]
        update.message.reply_text('Виберіть щось з
запропонованого...', reply_markup=ReplyKeyboardMarkup(
            reply_keyboard, one_time_keyboard=True))
    return ORDER
else:
    update.message.reply_text(
        'Наступного разу обирайте щось із запропонованого
натисніть /start')
    reply_keyboard = [

```

```

        ['🍲 Гарячі страви', '🥗 Холодні страви'], ['☕ Напої',
'🍰 Десерти']]
        update.message.reply_text('Все таки, що бажаєте
замовити?', reply_markup=ReplyKeyboardMarkup(
            reply_keyboard, one_time_keyboard=True,
input_field_placeholder=' Що бажаєте замовити?'
        ),)
        return ConversationHandler.END

def order(update: Update, context: CallbackContext) -> int:
    choice = update.message.text
    reply_keyboard = [['Так', 'Hi']]
    update.message.reply_text('Ви замовили ' + str(choice) +
'\nБажаєте ще щось замовити?',

reply_markup=ReplyKeyboardMarkup(reply_keyboard,
one_time_keyboard=True, input_field_placeholder='Бажаєте ще щось
замовити?'))
    my_order.append(str(choice))
    return DECISION

def decison(update: Update, context: CallbackContext) -> int:
    choice = update.message.text
    if choice == "Так

reply_keyboard = [['🍲 Гарячі страви', '🥗 Холодні страви'],
                    ['☕ Напої', '🍰 Десерти']]
        update.message.reply_text('Що ще бажаєте?',
reply_markup=ReplyKeyboardMarkup(
            reply_keyboard, one_time_keyboard=True,
input_field_placeholder=' Що бажаєте замовити?'
        ),
    )
        return MENU
    elif choice == "Hi":
        orders = ''
        value = 0
        # print(len(my_order))
        conn = sqlite3.connect('Randevu.db')
        cur = conn.cursor()
        cur.execute('PRAGMA foreign_keys = ON;')
        conn.commit()
        i = 3
        while i != len(my_order):
            cur.execute('Select Ціна from меню where Назва=?',
(my_order[i],))
            result = cur.fetchall()
            value = value + result[0][0]
            orders = orders + str(my_order[i]) + ', '
            i = i+1

```



```

update.message.reply_text(
    'Ваше замовлення прийнято, загальна 💰вартість: ' +
str(value) + 'грн.')
if len(str(my_order[0])) == 9:
    ans = " 📄ID: " + str(my_order[1])+" Столик: " + \
        str(my_order[2]) + '\n' + "Замовлення: " + orders
elif len(str(my_order[0])) == 10 or len(str(my_order[0]))
> 10:
    ans = " 📞Телефон: " + str(my_order[0])+" 📄ID: " +
str(
        my_order[1])+" Адреса: " + str(my_order[2]) + '\n'
+ "Замовлення: " + orders
else:
    print("err")
    # print(my_order)
    context.bot.send_message(chat_id, ans)
    return ConversationHandler.END

def cancel(update: Update, context: CallbackContext) -> int:
    user = update.message.from_user
    update.message.reply_text(
        'До зустрічі, надіюсь скоро зустрінемся. Для поновлення
нашого спілкування натисни /start',
        reply_markup=ReplyKeyboardRemove()
    )
    return ConversationHandler.END

def support_answer(update: Update, context: CallbackContext) ->
int:
    update.message.reply_text('Введіть відповідь на запитання:')
    return SUPPORT_ANSWER

def id_answer(update: Update, context: CallbackContext) -> int:
    global qanswer
    qanswer = update.message.text
    # print(qanswer)
    update.message.reply_text('Введіть ід користувача:')
    return ANSWER

def answer(update: Update, context: CallbackContext) -> int:
    chat_id = update.message.text
    # print(qanswer)
    # print(chat_id)
    try:
        context.bot.send_message(chat_id, qanswer)
        return ConversationHandler.END
    except:

```

```

        update.message.reply_text(
            'Нажаль користувач ' + str(chat_id) + ' заблокував
бота(')
        return ConversationHandler.END

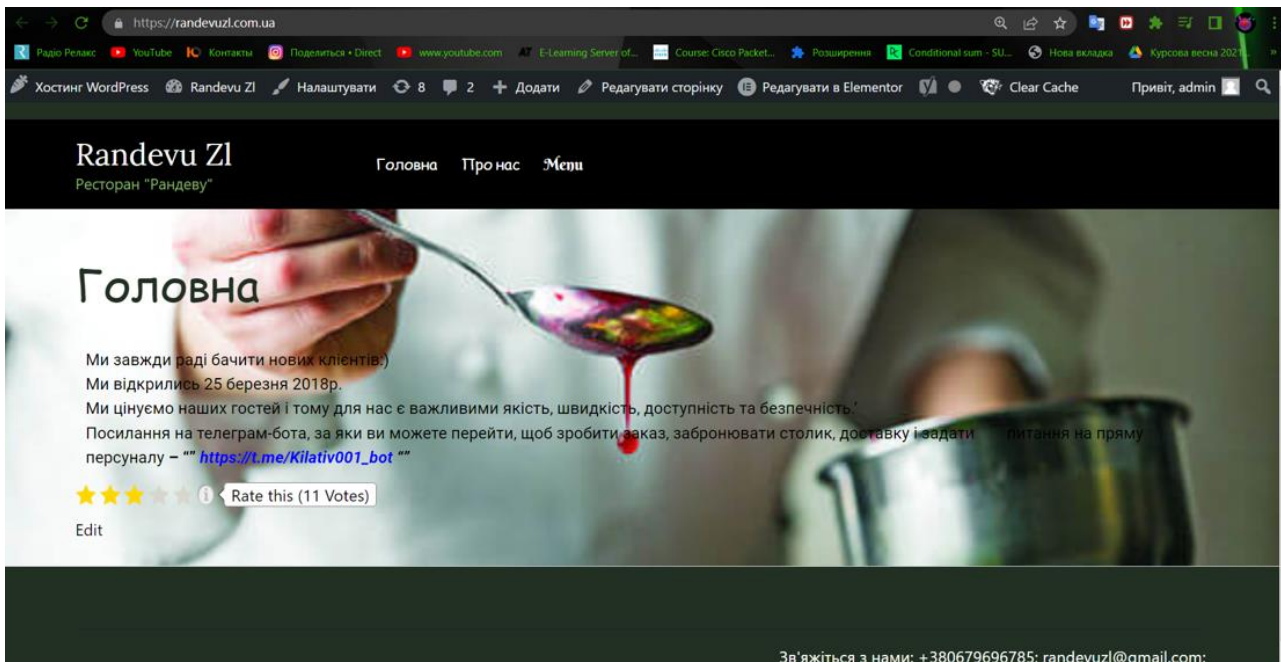
def main() -> None:
    updater = Updater(config.TOKEN)
    dispatcher = updater.dispatcher
    conv_handler = ConversationHandler(
        entry_points=[CommandHandler('start', start),
CommandHandler(
            'support_answer', support_answer)],
        states={
            CHECK: [MessageHandler(Filters.regex('^ (Віртуальний
офіціант|Доставка|Бронювання|Підтримка)$'), check)],
            TYPE: [MessageHandler(Filters.regex('^ (1|2|3|4|5)$'),
eat_type)],
            MENU: [MessageHandler(Filters.regex('^ (🍲 Гарячі
страви|🍓 Холодні страви|🍰 Десерти|🍷 Напої)$'), menu)],
            ORDER: [MessageHandler(Filters.text, order)],
            PHONENUMBER: [MessageHandler(Filters.regex('^ ([0-
9]{10})$'), getphonenumber)],
            DECISION: [MessageHandler(Filters.regex('^ (Так|Hi)$'),
decison)],
            ADDRESSA: [MessageHandler(Filters.text, getadressa)],
            SUPPORT: [MessageHandler(Filters.text, support)],
            SUPPORT_ANSWER: [MessageHandler(Filters.text,
id_answer)],
            ANSWER: [MessageHandler(Filters.text, answer)],

            DATE_BOOK: [MessageHandler(Filters.regex('^ (0[1-
9]|[12][0-9]|3[01])[-./](?:0[1-9]|1[012])[-./](?:19\d{2}|20[01][0-
9]|20[0-9]{2})$'), date_book)],
            TIME_BOOK: [MessageHandler(Filters.regex('^ ([01][0-
9]|2[0-3]):([0-5][0-9])$'), time_book)],
            CUSTOMER_COUNT: [MessageHandler(Filters.regex('^ ([1-
9]|\d{2,})$'), customer_count)],
            TABLE_BOOK:
[MessageHandler(Filters.regex('^ (1|2|3|4|5)$'), table_book)],
        },
        fallbacks=[CommandHandler('cancel', cancel)],
    )
    dispatcher.add_handler(conv_handler)
    updater.start_polling()
    updater.idle()

if __name__ == '__main__':

```

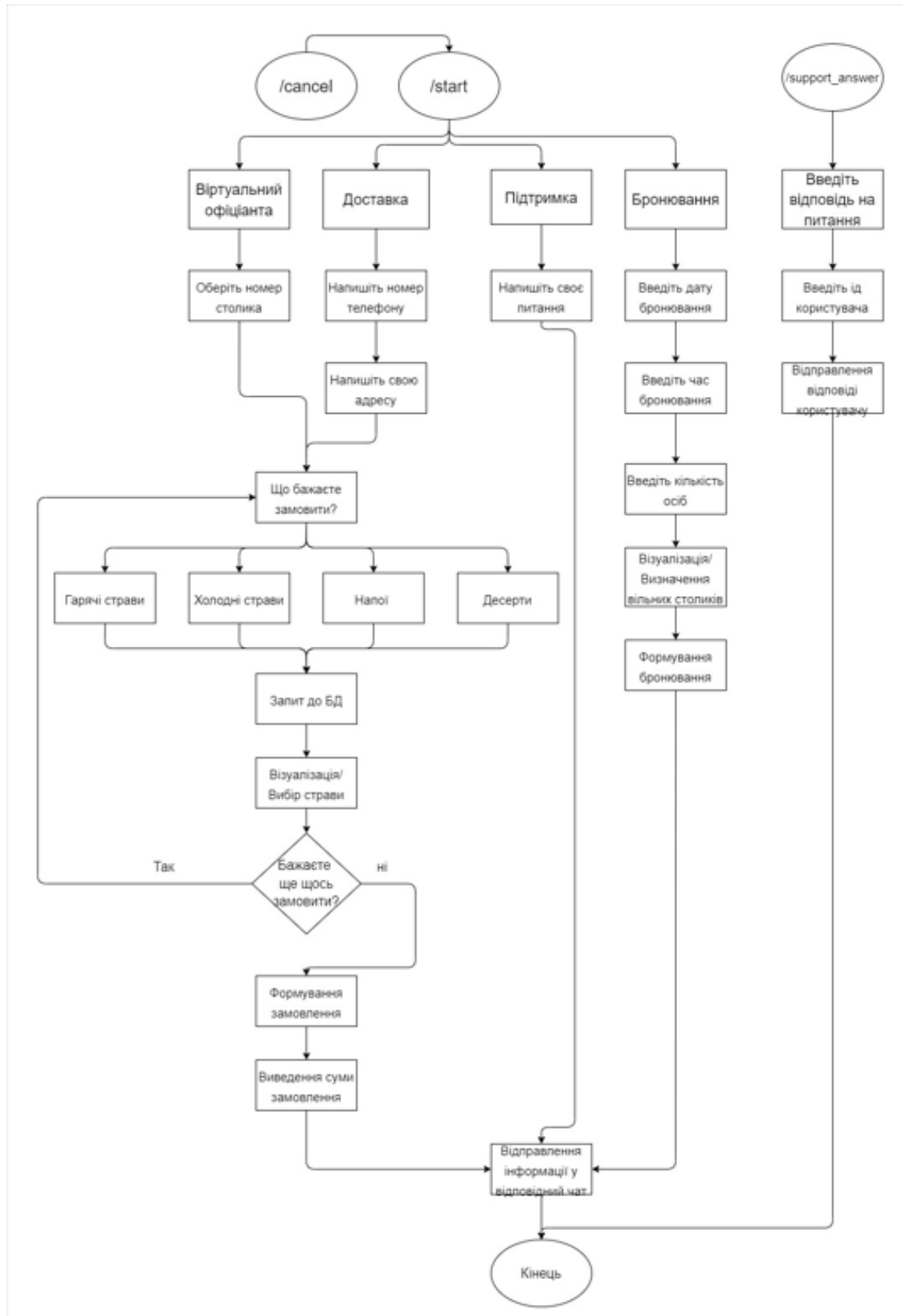
Головна сторінка сайту «Рандеву»



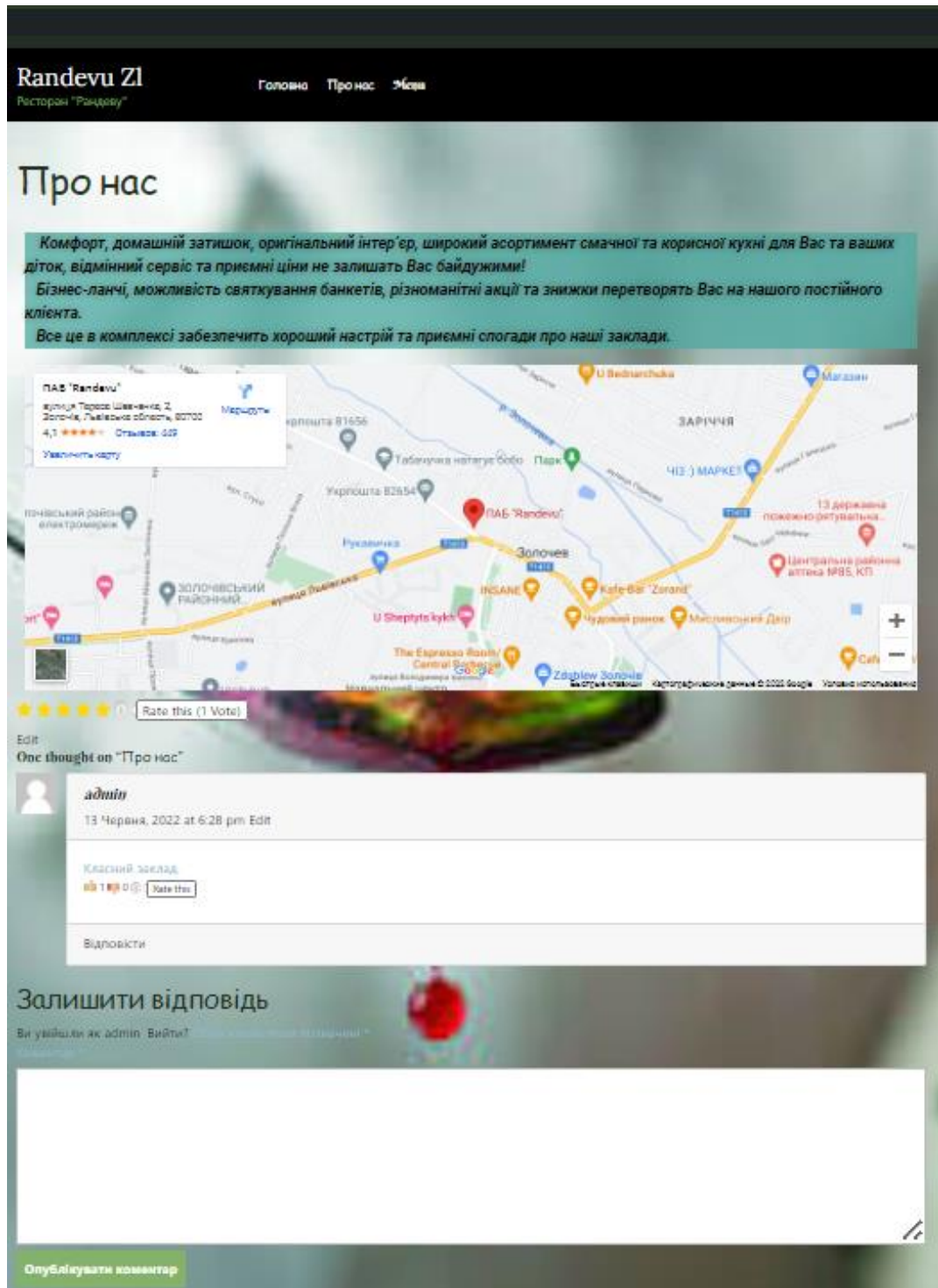
The screenshot shows the main page of the website 'Randevu Zl'. The browser address bar displays 'https://randevuzl.com.ua'. The website header includes the logo 'Randevu Zl' and the tagline 'Ресторан "Рандеву"'. The navigation menu contains 'Головна', 'Про нас', and 'Меню'. The main content area features a large image of a hand holding a spoon with a colorful liquid dripping from it. The text on the page reads: 'Головна', 'Ми завжди раді бачити нових клієнтів!', 'Ми відкрились 25 березня 2018р.', 'Ми цінуємо наших гостей і тому для нас є важливими якість, швидкість, доступність та безпеність.', and 'Посилання на телеграм-бота, за які ви можете перейти, щоб зробити заказ, забронювати столик, доставку і задати питання на пряму персоналу - "" https://t.me/Kilativ001_bot ""'. Below the text is a rating system with five stars and the text 'Rate this (11 Votes)'. An 'Edit' link is visible below the rating. At the bottom of the page, the contact information is provided: 'Зв'яжіться з нами: +380679696785; randevuzl@gmail.com;'. The browser's developer tools are open at the bottom of the page.

Додаток В

Блок-схема роботи телеграм-бота



Сторінка сайту про нас



Код оформлення меню

```
import sqlite3
conn = sqlite3.connect('Randevu.db')
cur = conn.cursor()
cur.execute("""CREATE TABLE IF NOT EXISTS меню(
    ід INT PRIMARY KEY,
    Назва TEXT,
    Тип TEXT,
    Інгредієнти TEXT,
    Ціна INT);
""")
cur.execute("""INSERT INTO меню(ід, Назва, Тип, Інгредієнти, Ціна)
VALUES('20', 'Плов', 'Гарячі страви', 'Мясо, зелень, приправи, рис', '150');""")
cur.execute("""INSERT INTO меню(ід, Назва, Тип, Інгредієнти, Ціна)
VALUES('21', 'Картошка пюре з відбивною', 'Гарячі страви', 'Свинина, Картопля, приправи',
'125');""")
cur.execute("""INSERT INTO меню(ід, Назва, Тип, Інгредієнти, Ціна)
VALUES('22', 'Борщ', 'Гарячі страви', 'Мясо, зелень, приправи, картопля, буряк', '190');""")
cur.execute("""INSERT INTO меню(ід, Назва, Тип, Інгредієнти, Ціна)
VALUES('23', 'Картошка фри', 'Гарячі страви', 'Картопля, приправи, рис', '100');""")
cur.execute("""INSERT INTO меню(ід, Назва, Тип, Інгредієнти, Ціна)
VALUES('24', 'Салат', 'Холодні страви', 'зелень, оливки, сир, соус', '75');""")
cur.execute("""INSERT INTO меню(ід, Назва, Тип, Інгредієнти, Ціна)
VALUES('25', 'Суші "Червони дракон"', 'Холодні страви', 'рис, тунець, сир маскарпоне, соус',
'150');""")
cur.execute("""INSERT INTO меню(ід, Назва, Тип, Інгредієнти, Ціна)
VALUES('26', 'Кава', 'Напої', 'Мелена кава', '30');""")
cur.execute("""INSERT INTO меню(ід, Назва, Тип, Інгредієнти, Ціна)
VALUES('27', 'Чорний Чай', 'Напої', 'Листочки чаю', '25');""")
cur.execute("""INSERT INTO меню(ід, Назва, Тип, Інгредієнти, Ціна)
VALUES('28', 'Фруктовий сік', 'Напої', 'Фрукти', '20');""")
cur.execute("""INSERT INTO меню(ід, Назва, Тип, Інгредієнти, Ціна)
VALUES('29', 'Мінералка', 'Напої', "", '15');""")
cur.execute("""INSERT INTO меню(ід, Назва, Тип, Інгредієнти, Ціна)
VALUES('30', 'Банановий пудинг із морозивом', 'Десерти', 'банан, морозиво, яйця, молоко',
'50');""")
cur.execute("""INSERT INTO меню(ід, Назва, Тип, Інгредієнти, Ціна)
VALUES('31', 'Нижний шоколадний фондан із кулькою морозива', 'Десерти', 'Чорний
шоколад, масло, яйця, морозиво', '50');""")
conn.commit()
```