МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій
Кафедра радіоелектронних і комп'ютерних систем

Допустити до захисту
Зав. кафедри
_____проф. Оленич І. Б.
_____
Призначити рецензентом
_____
Декан факультету
_____Фургала Ю. М.
_____

Кваліфікаційна робота
Бакалавр

"Розроблення інформаційної системи готелю з використанням мови Пайтон"

Дипломна робота спеціаліста
Група ФЕП-41
Виконавець
студент 4-го курсу
_____Горбаль Р. П.
_____
Науковий керівник
_____Флюнт О. Є.
_____

Львів – 2023

АНОТАЦІЯ

Метою даного бакалаврського проєкту є розробка інформаційної системи готелю, спрямованої на підвищення ефективності бізнес-процесу. Проєкт передбачає створення клієнт-серверної системи, де клієнтська частина представлена у вигляді веб-сайту, що надає потенційним клієнтам готелю доступ до функціональності системи. Розроблений сайт має сучасний адаптивний інтерфейс, що забезпечує зручну взаємодію користувачів з системою.

Серверна частина системи відповідає за обробку запитів від клієнтів, побудову веб-сторінок з використанням шаблонів, а також забезпечує взаємодію з базою даних. Ця система дозволяє ефективно вести бізнес та готельну діяльність, а також має потенціал для подальшого розширення та модифікації.

У процесі розробки проєкту були проаналізовані основні проблеми, пов'язані з впровадженням цифрових засобів у готельній галузі. Було досліджено наявні аналоги систем і обрано найбільш підходящі інструменти для розробки.

Однією з новизн у даній сфері є використання OpenAI API, який виступає у ролі розумного консультанта у системі. Цей інтелектуальний інтерфейс здатен надати відповіді на різноманітні запитання про діяльність готелю, що може бути цікавим для клієнтів.

Загалом, розробка цієї інформаційної системи готелю включає в себе використання сучасних технологій, урахування особливостей готельного бізнесу і впровадження інноваційних рішень для покращення якості обслуговування та ефективності діяльності готелю.

ABSTRACT

The goal of this bachelor's project is to develop a hotel information system aimed at increasing the efficiency of the business process. The project ensures the creation of a client-server system, where the client part is presented in the form of a website that provides hotel clients with access to the system's functionality. The developed site has a modern adaptive interface that ensures convenient interaction of users with the system.

The server part of the system is responsible for processing requests from clients, building web pages using templates, and also provides interaction with the database. This system enables efficient business and hotel operations and has the potential for further expansion and modification.

In the process of developing the project, the main problems related to the implementation of digital tools in the hotel industry were analyzed. Existing analogs of the system were investigated and the most suitable tools for development were selected.

One of the novelties in this area is the use of OpenAI API, which acts as a smart consultant in the system. This intelligent interface is capable of providing answers to various questions about hotel activities that may be of interest to customers.

In general, the development of this hotel information system includes the use of modern technologies, taking into account the peculiarities of the hotel business and the implementation of innovative solutions to improve the quality of service and the efficiency of the hotel.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....7
ВСТУП 8
РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ.....10
1.1 Сучасний стан та перспективи розвитку готельного господарства.....10
1.2 Аналіз існуючих на ринку рішень.....11
1.3 Формування конкретних задач та стандартів для розроблюваного продукту.....13
РОЗДІЛ 2. АНАЛІЗ ВИКОРИСТАНИХ ТЕХНОЛОГІЙ 19
2.1 Python фреймворки. Вибір доцільного фреймворку 19
2.2 Захист міжсайтових сценаріїв (XSS).....20
2.3 Шаблонізатор Jinja.....23
2.4 Аналіз та вибір СУБД.....25
2.5 Фреймворк Bootstrap для розробки веб-інтерфейсів.....27
2.6 Система управління версіями Git.....28
2.7 Принципи використання API Chat-GPT 29
РОЗДІЛ 3. ОПИС РОЗРОБЛЕНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ГОТЕЛЮ.31
3.1 Огляд функціональних можливостей та інтерфейсу системи готелю.....31
3.1.1 Клієнтська підсистема.....31
3.1.2 Серверна підсистема.....38
3.2 Шляхи подальшого потенційного розвитку системи.....43
ВИСНОВКИ.....44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....45
ДОДАТОК А. ОХОРОНА ПРАЦІ ТА ІНФОРМАЦІЙНІ СИСТЕМИ ТУРИСТИЧНОГО КОМПЛЕКСУ 48
ДОДАТОК Б. ЛІСТИНГ ПРОГРАМИ.....51

ПЕРЕЛІК СКОРОЧЕНЬ

БД	База даних
СУБД	Система управління базами даних.
API	Application Programming Interface (укр. Прикладний програмний інтерфейс) – це набір чітко визначених методів для взаємодії різних компонентів програми.
CSRF	Cross-Site Request Forgery (укр. Міжсайтова підробка запиту) -тип атаки на веб-ресурс.
GPT	Generative Pre-trained Transformer (укр. Генеративний попередньо навчений трансформатор) – породжувальне попереднє тренування.
IDE	Integrated Development Environment (укр. Інтегроване середовище розробки) – комплекс програмних засобів, використовуваний програмістами розробки програмного обеспечения.
ORM	Object-Relational Mapping (укр. Об’єктно-реляційне відображення) – технологія програмування, яка пов’язує бази даних із концепціями об’єктно-орієнтованих мов програмування
SQL	Structured query language (укр. Мова структурованих запитів).
URL	Uniform Resource Locator (укр. Уніфікований локатор ресурсів) – стандартизована адреса ресурсу в інтернеті.
XSS	Cross-site scripting (укр. Міжсайтовий скриптинг) – тип атаки на веб-ресурс.

ВСТУП

У сучасному світі, де гостинність та якісне обслуговування мають велике значення для успіху готелю, інформаційні системи стають невід'ємною складовою частиною готельного бізнесу. Однак, зростання конкуренції та зміна уявлень клієнтів щодо якості обслуговування ставлять перед готелями нові виклики. Виникає потреба в ефективних та інноваційних рішеннях для автоматизації робочих процесів та покращення клієнтського досвіду.

Метою даного бакалаврського проєкту є розробка інформаційної системи готелю, що спрямована на підвищення ефективності бізнес-процесу. Розробка включає створення клієнт-серверної системи, де клієнтська частина представлена у вигляді сучасного адаптивного веб-сайту, призначеного для потенційних клієнтів готелю. Система надає різноманітні можливості, що сприяють зручній взаємодії з готелем. Розробка інформаційної системи для готелю з використанням мови Python та інтеграцією API Chat-GPT є основною метою, оскільки це впровадить автоматизацію роботи клієнтської підтримки. Важливим завданням роботи є створення функціонального, зручного у використанні та ефективного інструменту, який допоможе готелю покращити якість обслуговування, забезпечити швидкий та точний відгук на запитання клієнтів, а також оптимізувати роботу персоналу готелю.

Серверна частина системи відповідає за обробку запитів від клієнтів, побудову сторінок веб-сайту з використанням шаблонів та взаємодію з базою даних. Це забезпечує швидку та надійну обробку інформації, а також забезпечує зручність в роботі для адміністраторів готелю.

Об'єктом дослідження є інформаційна система готелю з використанням мови Python та API Chat-GPT для автоматизації роботи клієнтської підтримки. Дослідження спрямоване на вивчення та аналіз можливостей використання цих технологій для покращення ефективності та якості обслуговування в готелі. Цей інтелектуальний інтерфейс дозволяє автоматично відповідати на запитання клієнтів, надаючи їм швидку та точну інформацію. Це сприяє зниженню навантаження на персонал готелю та поліпшенню якості обслуговування.

Предметом дослідження є функціональні можливості системи. Головний акцент роботи скерований на розробку модулів, що дозволять автоматизувати обробку запитань від клієнтів, генерацію адекватних відповідей та оптимізацію процесу взаємодії з клієнтами готелю.

Для досягнення поставлених цілей у роботі будуть використані методи аналізу літературних джерел, аналізу вимог та процесів готельного бізнесу, програмування з використанням мови Python, розробки веб-додатків та інтеграції API. Для реалізації проекту використовуватимуться сучасні розробницькі інструменти та середовища, такі як Python-фреймворк Django, IDE Pycharm, веб-сервери та бази даних sqlite.

Розробка інформаційної системи готелю з API Chat-GPT для автоматизації роботи клієнтської підтримки є інноваційним підходом, який може принести значні переваги готельному бізнесу. Результати дослідження можуть бути застосовані в готелях різного розміру та рівня обслуговування, що дозволить покращити якість комунікації з клієнтами, зменшити час очікування на відповідь та підвищити загальну задоволеність клієнтів.

Після реалізації проєкту, можливі додаткові напрями розвитку. Зокрема, можливо розширення функціональності системи, наприклад, додавання інтеграції з іншими сервісами та API, розробка мобільного додатку для зручного доступу до системи, а також вдосконалення алгоритмів обробки запитань та генерації відповідей з використанням штучного інтелекту та машинного навчання.

Отже, цей проєкт пропонує інноваційні рішення для готельного бізнесу та може стати цінним інструментом для покращення якості обслуговування та задоволення потреб клієнтів. Розроблена інформаційна система має потенціал покращити якість обслуговування, зменшити ресурсомісткість процесів та забезпечити конкурентну перевагу готелю на ринку гостьових послуг.

РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

1.1 Сучасний стан та перспективи розвитку готельного господарства

Сучасне готельне господарство України переживає період активного розвитку та змін. Готельна галузь стикається з рядом викликів і проблем. Однією з найбільш важливих проблем є недостатній рівень сервісу та якості обслуговування в багатьох готелях. Це може бути пов’язано зі старінням готельного парку, недостатньою кваліфікацією персоналу або недостатньою увагою до деталей.

Окрім того, відсутність ефективної системи взаємодії з клієнтами є ще однією проблемою. Клієнти очікують швидкого та зручного доступу до інформації про готелі, можливості онлайн-бронювання та швидкого реагування на їх запити та заявки. Недолік автоматизованих систем підтримки та відсутність інтелектуальних рішень призводять до неефективного використання ресурсів та неситуаційного обслуговування клієнтів.

Однак, в перспективі готельне господарство України має значний потенціал для подальшого розвитку. Зростання туристичного попиту, впровадження новітніх технологій та покращення якості обслуговування можуть сприяти залученню більшої кількості туристів та підвищенню рівня задоволеності клієнтів.

Одним з потенційних напрямів розвитку є використання інформаційних систем, які автоматизують процеси обробки запитань клієнтів та підтримки. Розробка і впровадження інформаційної системи готелю з використанням мови Python [1] та API Chat-GPT [2] може стати важливим кроком у цьому напрямку. Це дозволить покращити якість обслуговування, зменшити час відповіді на запити клієнтів та забезпечити більш ефективну взаємодію з ними.

Використання такої системи може сприяти підвищенню конкурентоспроможності готелів, залученню нових клієнтів та формуванню позитивного іміджу країни в цілому. Розвиток готельного господарства України вимагає постійного вдосконалення та впровадження нових технологій, що

забезпечать більш зручну та задоволену взаємодію між готелями та їх клієнтами.

1.2 Аналіз існуючих на ринку рішень

Стан проблемної області з розробки інформац ійних систем для готелів, вимагає уваги та вдосконалення. На сьогоднішній день, багато готелів все ще мають обмежені можливості в автоматизації та оптимізації своєї роботи, зокрема в сфері клієнтської підтримки.

Одна з ключових проблем полягає в неефективності обробки запитань та відповідей між клієнтами та готелем. Клієнти часто зіштовхуються з довгими чергами на телефонну підтримку, або затримками у відповідях на електронні листи, що призводить до незадоволення та втрати потенційних клієнтів.

Більш того, багато готелів не мають ефективних систем бронювання та інформаційних порталів, що ускладнює процес пошуку наявних номерів, порівняння цін та здійснення бронювання. Клієнти зацікавлені в швидкій та зручній процедурі бронювання, але не завжди можуть знайти потрібну інформацію легко та швидко.

Крім того, зростаючий обсяг даних, що зберігаються та обробляються готелями, вимагає впровадження надійних механізмів безпеки та захисту конфіденційної інформації клієнтів. Врахування вимог GDPR [3] та інших регуляторних стандартів стає особливо важливим для готелів, оскільки порушення конфіденційності може призвести до серйозних правових наслідків та втрати довіри клієнтів.

Сучасні технології та інтернет-ресурси змінюють підходи до ведення готельного бізнесу. Віртуальні платформи для пошуку та бронювання готелів стали невід’ємною частиною життя багатьох мандрівників і туристів. Зростаюча конкуренція та зміни у споживацькому підході вимагають від готелів активного використання інформаційних технологій для покращення якості обслуговування та залучення більшої кількості клієнтів.

Значний ріст технологій і програмного забезпечення для готельної галузі створив широкий вибір існуючих рішень на ринку. При виборі інформаційної системи, необхідно враховувати декілька основних факторів:

- Функціональність;
- Інтеграція;
- Масштабованість;
- Безпека;
- Вартість.

Важливо оцінити, які конкретні функції та можливості надають існуючі системи. Це можуть бути бронювання онлайн, управління клієнтськими даними, фінансовий облік, звітність тощо. Потрібно зрозуміти, чи відповідає ця функціональність конкретним потребам. Також важливо з'ясувати, як добре система може інтегруватися з іншими системами, які вже використовуються або плануються використовуватись. Наприклад, інтеграція з системами управління бронюванням, платіжними шлюзами, фінансовими системами та іншими може спростити бізнес-процеси. Якщо готель має потенціал для розширення, важливо вибирати систему, яка зможе витримати зростання бізнесу. Варто також зазначити, що збереження даних клієнтів та бізнес-інформації є критично важливою задачею для готельних систем. Потрібно неодмінно, переконатись, що існуючі рішення мають високі стандарти безпеки, включаючи шифрування даних, захист від несанкціонованого доступу та резервне копіювання даних.

Перед розробкою інформаційної системи для готелю з використанням мови Python, необхідно провести аналіз існуючих рішень на ринку. Давайте розглянемо основні з них:

1. Property Management Systems (PMS) [4] : Ці системи спеціалізуються на управлінні готелями, включаючи бронювання, облік клієнтів, фінансовий облік та звітність. Вони зазвичай мають веб-інтерфейс і базуються на різних технологіях програмування. Деякі PMS надають API для інтеграції з іншими системами.

2. Online Booking Engines [5] : Це системи, які забезпечують можливість онлайн-бронювання готельних номерів. Вони надають інтерфейс для клієнтів для пошуку наявних номерів, порівняння цін та здійснення бронювання. Багато з них також підтримують інтеграцію з платіжними шлюзами.
3. Customer Relationship Management (CRM) Systems [6] : Ці системи дозволяють збирати, зберігати та аналізувати дані про клієнтів готелю. Вони надають інструменти для взаємодії з клієнтами, управління контактами, маркетингу та звітності.
4. Channel Managers [7] : Ці системи допомагають готелям керувати своєю наявністю та цінами на різних онлайн-каналах бронювання, таких як OTA (Online Travel Agency) та мережі готелів. Вони автоматизують процес синхронізації даних про наявність номерів та ціни між різними платформами.
5. Chatbot Systems [8] : Ці системи використовують штучний інтелект для автоматизації клієнтської підтримки через чат-інтерфейс. Вони можуть відповідати на запитання клієнтів, надавати інформацію про готель, процес бронювання та інше.

Комбінація цих рішень може створити інформаційну систему готелю, яка забезпечує ефективне управління бізнес-процесами, зручне онлайн-бронювання для клієнтів і автоматизовану підтримку через чат-інтерфейс. Вибір конкретних рішень буде залежати від потреб готелю, бюджету та інших факторів.

1.3 Формування конкретних задач та стандартів для розроблюваного продукту

Впровадження інформаційної системи готелю має свої плюси і мінуси, які варто врахувати при оцінці його ефективності:

Переваги:

Автоматизація та оптимізація: Впровадження інформаційної системи дозволяє автоматизувати багато рутинних процесів, таких як облік клієнтів,

атоматизована клієнтська підтримка та звітність. Це сприяє підвищенню ефективності роботи готелю, зменшенню помилок та оптимізації використання ресурсів.

Зручність для клієнтів: Завдяки розробленій клієнтській частині, яка включає адаптивний інтерфейс та інтеграція з Chat GPT, клієнти можуть легко шукати наявні номери, порівнювати ціни та писати коментарі, на підставі цих даних. Це покращує їх досвід користування та забезпечує зручність взаємодії з системою.

Покращена клієнтська підтримка: Інтеграція з API Chat-GPT дозволяє автоматизувати роботу клієнтської підтримки, надаючи клієнтам швидкі та точні відповіді на запитання про готель, його основних характеристик та інших послуг. Це сприяє поліпшенню обслуговування клієнтів та збільшенню задоволеності. Для власників готелів ця технологія може стати у нагоді, оскільки вона зменшує фінансові витрати на працівників клієнтської підтримки і надає “роздинки” їхньому сайту. Системи з використанням цієї технології неодмінно будуть у тренді, оскільки інтеграція з API Chat-GPT – новизна у готельному бізнесі і такий підхід у сучасних реаліях неодмінно буде “іти у ногу з часом ”, що забезпечить прихід клієнтів.

Збереження та аналіз даних: Інформаційна система дозволяє збирати, зберігати та аналізувати дані, що надає клієнт. Це надає готелю цінні інсайти для прийняття стратегічних рішень, управління маркетингом та покращення якості обслуговування.

Недоліки:

Складність впровадження: Розробка та впровадження комплексної інформаційної системи може бути складним процесом, який вимагає часу та зусиль з боку команди розробників. Необхідно гарантувати якість, забезпечити безпеку та правильну інтеграцію з існуючими системами.

Ризик технічних проблем: Інформаційна система може стикатися з технічними проблемами, такими як збої в роботі серверів, проблеми з

мережевим з’єднанням або безпекові загрози. Потрібно мати плани відновлення та забезпечення безпеки, щоб зменшити ризики.

Враховуючи переваги та недоліки, важливо провести аналіз витрат і очікуваних користей для прийняття рішення про впровадження інформаційної системи готелю. Зробивши це, можна визначити, чи відповідає цей підхід потребам готелю та його клієнтів.

При розробці інформаційної системи для готелю варто врахувати різні аспекти, які сприятимуть успішному впровадженню проекту.

Спочатку необхідно зосередитись на розробці клієнтської частини, яка включає веб-сайт готелю. Основною метою створення інтерфейсу, що дозволить потенційним клієнтам отримувати інформацію про готель та його послуги.

Паралельно з розробкою клієнтської частини, важливо зосередитись на створенні серверної частини системи. Потрібно розробити сервер, який зможе обробляти запити від клієнтів та забезпечувати необхідний функціонал системи. Сторінки сайту необхідно побудувати з шаблонів, щоб зручно відображати інформацію про готель, наявні номери, ціни та інші дані. Також забезпечте можливість обробки форм та взаємодії з базою даних для збереження та отримання інформації від клієнтів.

Ще одним важливим аспектом є інтеграція з API Chat-GPT, що дозволить автоматизувати роботу клієнтської підтримки. У веб-системі необхідно забезпечити можливість відповідати на запитання клієнтів та надавати інформацію про готель та інші необхідні функції підтримки.

При розробці системи також варто приділити увагу ефективності та масштабованості. Потрбіно створити оптимізовану архітектуру системи, яка забезпечить швидку обробку запитів та високу продуктивність. Використовувати кращі практики розробки, такі як кешування та оптимізацію запитів до бази даних, для забезпечення ефективності системи. Додатково передбачити можливість масштабування системи для підтримки зростаючої кількості клієнтів та оптимального розширення функціоналу.

Реалізація тестування системи допоможе виявити та усунути помилки та проблеми. Використання рекомендованих стандартів програмування та архітектури лише забезпечить якість коду та полегшить його супровід.

При формуванні конкретних задач і стандартів варто враховувати особливості системи, вимоги клієнтів, а також наявні ресурси.

SWOT-аналіз [9], що наведений на Рис. 1.1, допоможе визначити переваги, недоліки, можливості та загрози, що впливають на інформаційну систему готелю, та розробити стратегію розвитку з вдосконалення системи на основі вище наведених аспектів. Стратегія розвитку має враховувати сильні сторони системи, вирішення слабких сторін та використання можливостей для покращення обслуговування клієнтів, збільшення ефективності та зниження ризиків. Реалізація цієї стратегії допоможе зміцнити позиції готелю на ринку та задовольнити потреби клієнтів.

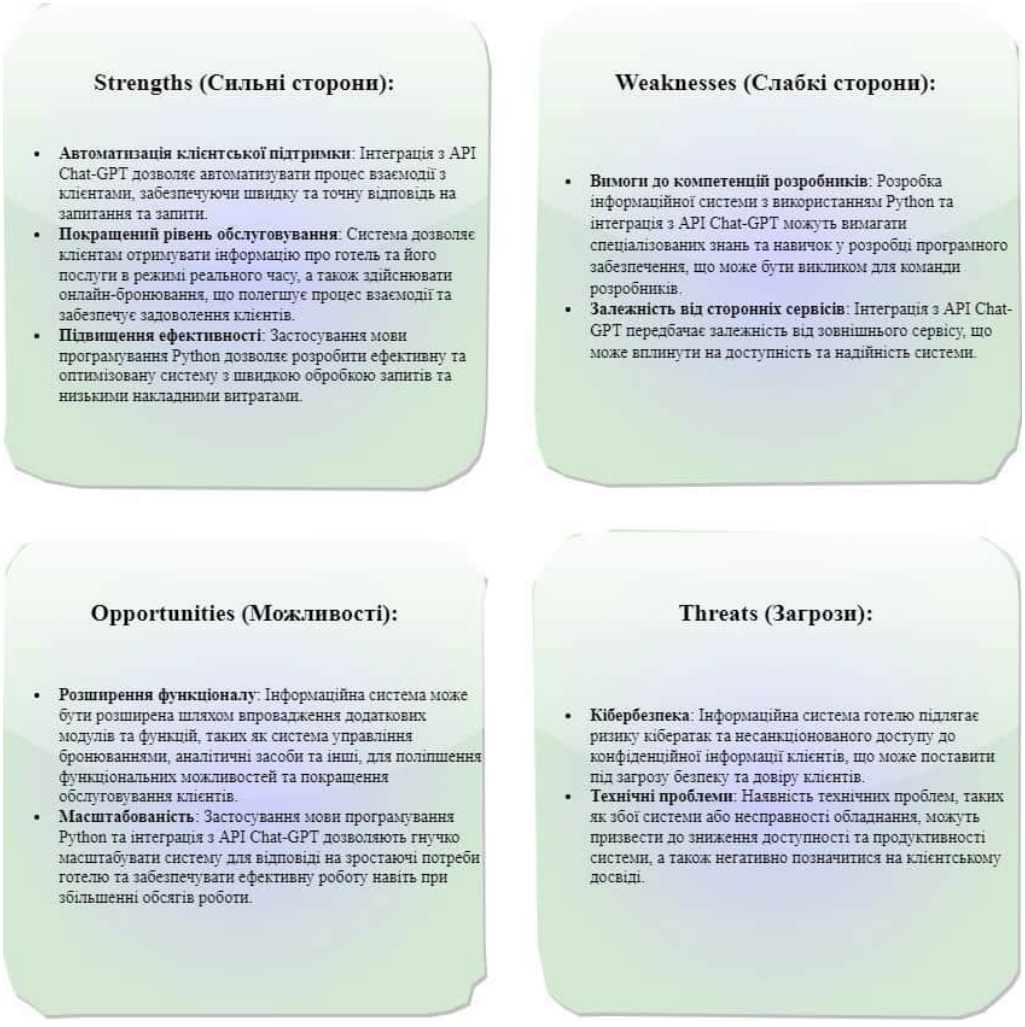

Рис. 1.1. SWOT-аналіз інформаційної системи готелю

SWOT-аналіз інформаційної системи готелю надав важливу інформацію про переваги, недоліки, можливості та загрози, що впливають на систему. На основі аналізу можна зробити наступні висновки:

Сильні сторони, такі як автоматизація клієнтської підтримки, покращений рівень обслуговування та підвищення ефективності, створюють переваги для готелю, забезпечуючи швидку та точну відповідь на запитання клієнтів та полегшуючи їх взаємодію з готелем.

Серед слабких сторін можна виділити вимоги до компетенцій розробників та залежність від сторонніх сервісів. Ці фактори можуть створювати труднощі при розробці та підтримці системи, але можуть бути подолані через надання необхідних навчань та пошук альтернативних рішень.

Можливості, такі як розширення функціоналу та масштабованість системи, відкривають шляхи для покращення та розвитку інформаційної системи готелю, що сприятиме поліпшенню обслуговування клієнтів та забезпеченню ефективної роботи.

Загрози, такі як кібербезпека та технічні проблеми, потребують уваги та заходів безпеки для запобігання можливим проблемам та забезпечення захисту конфіденційної інформації клієнтів.

Отже, на основі SWOT-аналізу інформаційної системи готелю можна сформувати такі конкретні задачі та стандарти:

Автоматизації клієнтської підтримки: Враховуючи сильну сторону, можна зосередитися на вдосконаленні системи відповіді на запитання та запити клієнтів. Забезпечення швидкості та точності відповідей шляхом покращення моделей Chat-GPT або впровадження додаткових алгоритмів машинного навчання може покращити якість обслуговування.

Збільшення функціоналу системи: Використовуючи можливості розширення функціоналу, можна впровадити додаткові модулі та функції, що поліпшать обслуговування клієнтів. Наприклад, розробка системи аналітичних засобів допоможе покращити аналізу даних.

Зосередження на масштабованості: Завдяки можливості масштабування системи на основі Python та API Chat-GPT, слід розглянути планування розширення та підготовку системи до зростання обсягів роботи. Це може включати оптимізацію інфраструктури, горизонтальне масштабування та використання хмарних рішень для забезпечення ефективності та продуктивності системи.

РОЗДІЛ 2. АНАЛІЗ ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

2.1 Python фреймворки. Вибір доцільного фреймворку

Згідно з даними опитування розробників Python у 2019 [10], Django [11] та Flask [12] є найпопулярнішими веб фреймворками. Ви навряд чи помилитеся, вибравши один із цих фреймворків для роботи з вашим новим веб додатком. При виборі того, який з них краще працюватиме, є ряд явних відмінностей, які потрібно мати на увазі перед тим, як зробити вибір.

Незалежно від того, який фреймворк буде використовуватись, вони обидва добре задокументовані і широко використовуються, так що можна не турбуватися про наявність підтримки з боку спільнот. Отже, без зайвих слів, проаналізуємо те, що пропонує кожен фреймворк і окреслимо відмінності та спільні риси.

Flask і Django надають аналогічний функціонал в обробці запитів, підтримці документів, але різняться в масштабі відповідальності. Більшість відмінностей між двома фреймворками випливають із різних підходів, решта – із відмінних основних проектних рішень.

Форми: Django доступний із вбудованими формами, які інтегруються з ORM та адмінкою сайту. Flask не підтримує форми за замовчуванням, але можна використовувати WTForms [13], щоб заповнити цю прогалину.

Бази даних: Django доступний із вбудованою ORM та системою міграції [14], яка може керувати базами даних. Flask не може цим похвалитися, однак є такі інструменти, як SQLAlchemy [15], які надають аналогічний функціонал (або навіть більше).

Аутентифікація та привілеї користувачам: Django надає програму автентифікації, яка надає реалізацію за умовчанням для керування користувача та привілеїв. Flask надає безпечні куки як інструмент вашої власної реалізації.

Панель адміністратора: Django включає повністю інтегрований адмін-інтерфейс для управління даними програми. Flask не має таких функцій, але Flask-Admin – дуже популярне розширення, яке можна використовувати для створення аналогічного адміністративного інструменту.

На офіційному сайті Django [16], що відображений на Рис. 2.1 стверджують, що Django спрощує створення веб-додатків, робить це швидше, використовуючи менше коду і називають Django “веб-фреймворк для перфекціоністів з дедлайнами”. Справді, Django – це хороший фреймворк, який (за замовчуванням) робить безліч рішень самостійно, так що користувач має інструмент для створення типових веб додатків.

Отже, для створення додатку, доцільним буде обрати саме фреймворк Django.

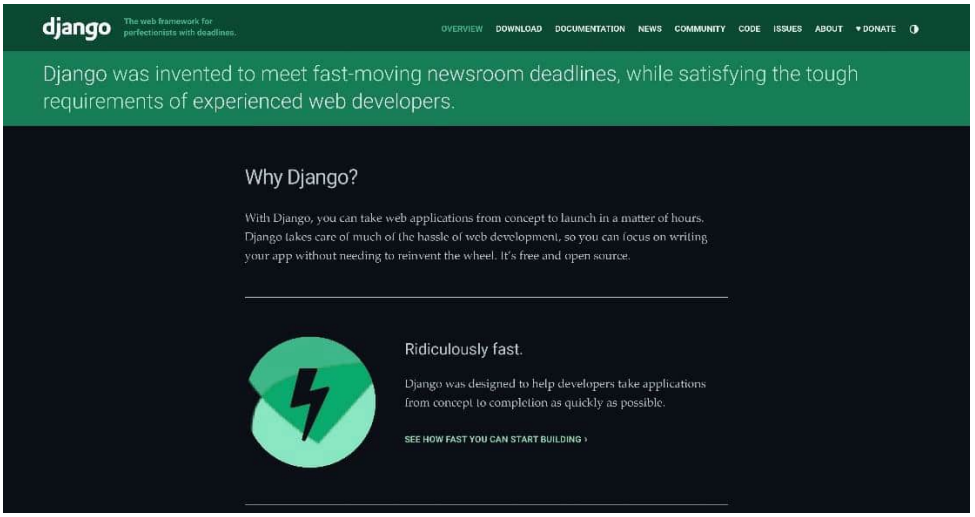

Рис. 2.1. Головна сторінка фреймворку Django

2.2 Захист міжсайтових сценаріїв (XSS)

XSS-атаки [17] дозволяють користувачеві впроваджувати сценарії на стороні клієнта в браузери інших користувачів. Зазвичай це досягається шляхом зберігання зловмисних сценаріїв у базі даних, де вони будуть отримані та відображені іншим користувачам, або шляхом спонукання користувачів клацати посилання, яке призведе до виконання JavaScript зловмисника у веб-переглядачі користувача. Однак атаки XSS можуть виникати з будь-якого ненадійного джерела даних, наприклад файлів cookie або веб-служб, якщо дані недостатньо оброблені перед додаванням на сторінку.

Використання шаблонів Django захищає вас від більшості атак XSS. Однак важливо розуміти, який захист він забезпечує та його обмеження.

Атаки CSRF дозволяють зловмисникам виконувати дії з використанням облікових даних іншого користувача без відома або згоди цього користувача (Див. Рис. 2.2).

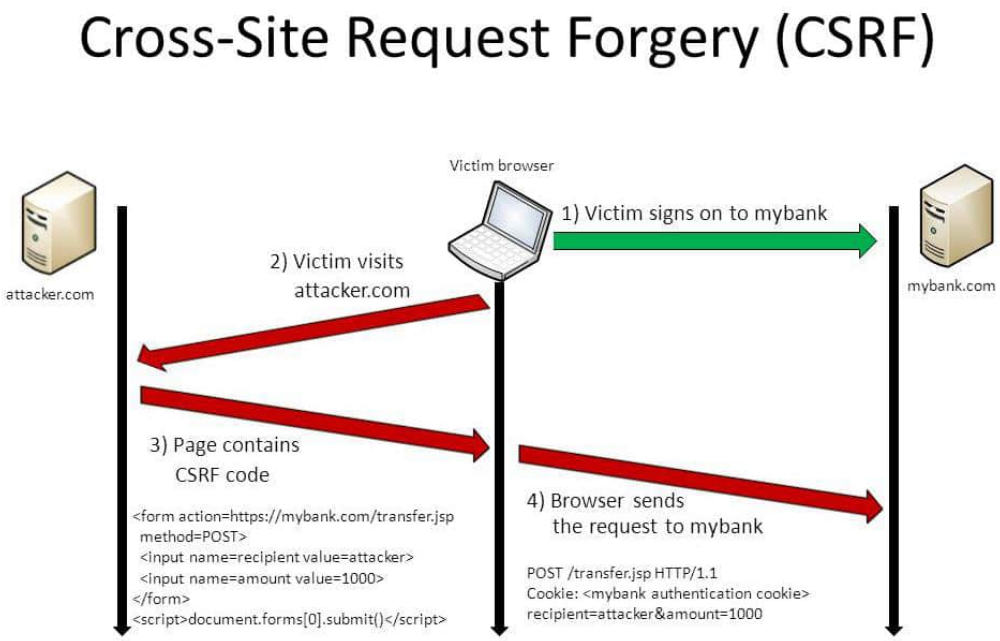

Рис. 2.2. Графічне представлення перехоплення CSRF токена

Django має вбудований захист від більшості типів CSRF-атак, за умови, що ви ввімкнули та використовували його, де це необхідно. Однак, як і у будь-якої техніки пом’якшення, існують обмеження. Наприклад, можна вимкнути модуль CSRF глобально або для окремих переглядів. Ви повинні робити це, лише якщо знаєте, що робите. Існують інші обмеження, якщо ваш сайт має субдомени, які ви не контролюєте.

Захист CSRF працює шляхом перевірки секрету в кожному запиті POST. Це гарантує, що зловмисний користувач не зможе “повторно відтворити” форму POST на веб-сайті та змусити іншого користувача мимоволі надіслати цю форму. Зловмисник мав би знати секрет, який залежить від користувача (за допомогою файлів cookie).

Набори запитів Django також захищені від SQL-ін’єкцій [18], оскільки їх запити створюються за допомогою параметризації запитів. SQL-код запиту визначається окремо від параметрів запиту. Оскільки параметри можуть бути надані користувачем і, отже, небезпечні, вони екрануються базовим драйвером

бази даних. Принцип роботи SQL-ін’єкції зображено на Рис. 2.3.

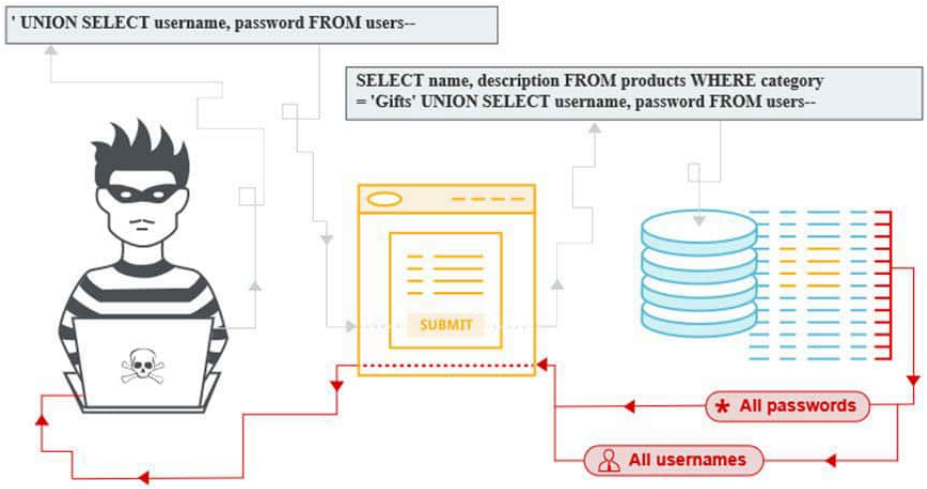

Рис. 2.3. Графічне представлення застосування SQL-ін’єкції

Також Django містить захист від клікджекінгу [19] у формі проміжного програмного забезпечення X-Frame-Options [20], яке у підтримуваному браузері може запобігти візуалізації сайту у фреймі. Можна вимкнути захист для кожного перегляду або налаштувати точне значення заголовка, що надсилається. Принцип роботи клікджекінгу зображено на Рис. 2.4.

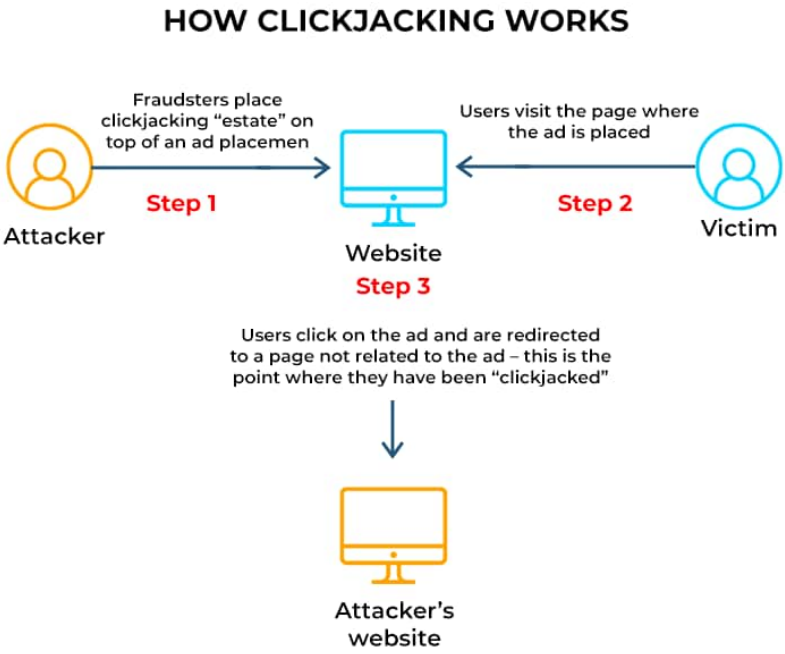

Рис. 2.4. Графічне представлення застосування клікджекінгу

Отже, фреймворк має вбудовані засоби захисту від різних типів атак, таких як XSS, CSRF і SQL-ін’єкції. Використання шаблонів Django допомагає у запобіганні більшості XSS-атак. Захист CSRF вбудований у Django і допомагає у запобіганні недозволеним діям з використанням облікових даних користувачів. Django також має механізми для захисту від SQL-ін’єкцій, шляхом параметризації запитів і екранування небезпечних параметрів. Проте, важливо знати обмеження цих заходів безпеки і правильно їх конфігурувати та використовувати. Також, Django надає можливості для написання власних SQL-запитів, але це потребує обережного використання, щоб уникнути вразливостей. Захист від клікджекінгу також включений у Django за допомогою форми проміжного програмного забезпечення X-Frame-Options.

Враховуючи ці функціональні можливості та їх обмеження, можна зробити висновок, що використання Django може ефективно захистити додаток від різних атак та забезпечити безпеку користувачів.

2.3 Шаблонізатор Jinja

Jinja [21] є популярним движком шаблонів для мови програмування Python і широко використовується в фреймворках, включаючи Django. Використання Jinja у Django дозволяє розробникам створювати потужні і гнучкі шаблони для веб-сторінок.

Основним призначенням Jinja у Django є розмежування логіки бізнес-логіки і представлення даних у шаблонах (Див. Рис. 2.5). Завдяки синтаксису Jinja, розробники можуть легко виконувати операції, обробляти дані і управляти потоком шаблону.

Jinja надає простий і зрозумілий синтаксис, який дозволяє розробникам легко створювати шаблони без зайвої складності. За допомогою Jinja можна використовувати умовні оператори і цикли для динамічного відображення даних у шаблонах. Це дозволяє створювати більш гнучкі і адаптивні шаблони, як показано на Рис. 2.5.

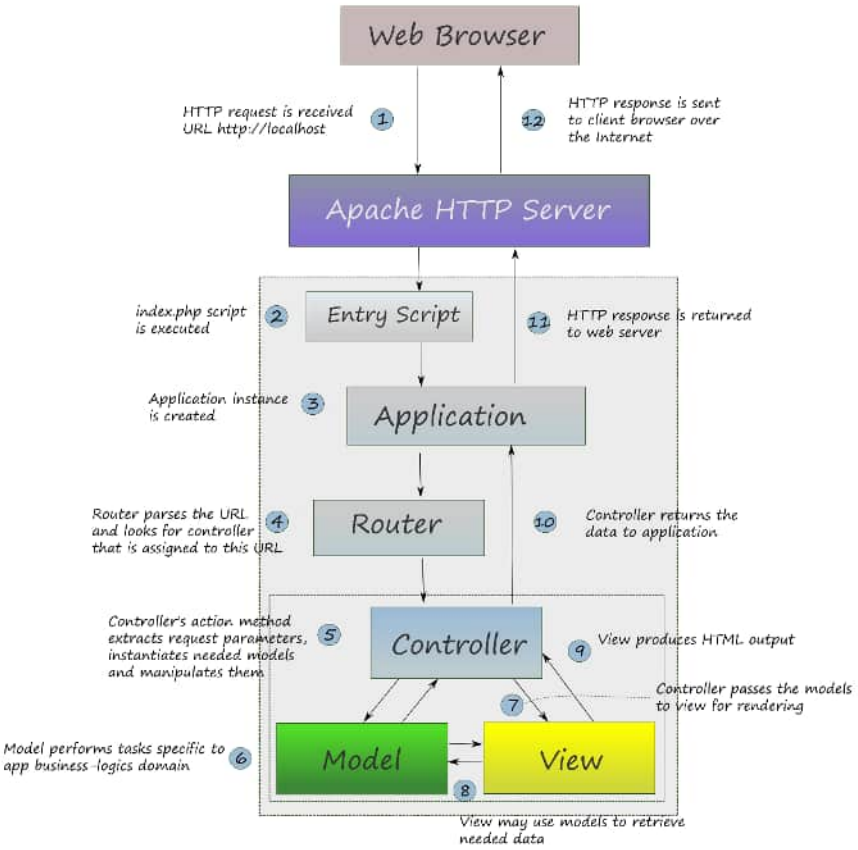

Рис. 2.5. Розділення бізнес-логіки та її представлення

Також Jinja підтримує механізм наслідування шаблонів, що дозволяє створювати базові шаблони зі спільними елементами та наслідувати їх для створення специфічних шаблонів для окремих сторінок або компонентів.

Шаблонізатор надає широкий набір вбудованих фільтрів і можливість створення власних фільтрів для обробки даних у шаблонах. Також є можливість визначати макроси для повторного використання складних фрагментів коду.

Використання Jinja у Django допомагає запобігати XSS-атакам, шляхом автоматично екранування потенційно небезпечних даних, перед тим як вони відображаються у шаблонах.

Застосування Jinja у Django дозволяє розробникам ефективно відокремлювати логіку від представлення і створювати чистий, зрозумілий і повторно використовуваний код шаблонів. Він доповнює можливості фреймворка Django і забезпечує розширення та гнучкість у розробці веб-

додатків, саме тому застосування цього шаблонізатора буде доцільним для проекту.

2.4. Аналіз та вибір СУБД

Система управління базами даних (СУБД) є набором програм, призначених для збору, оптимізації, адміністрування та організації баз даних. Вона виконує завдання створення та зберігання бази даних, а також керування нею для коригування або зміни даних. СУБД можуть бути різними, залежно від моделі даних чи виду доступу. Деякі з них включають мережеві, реляційні, об’єктно-орієнтовані, клієнт-серверні, файл-серверні та вбудовані.

Розглянемо декілька з них, а саме: MySQL [22], PostgreSQL [23] та SQLite [24].

Рис. 2.6. Головна сторінка СУБД MySQL

MySQL є однією з найпопулярніших відкритих реляційних СУБД (Головна сторінка представлена на Рис. 2.6). Вона використовує мову запитів SQL для взаємодії з базою даних. MySQL пропонує широкий набір функцій, високу продуктивність та надійність. Вона широко використовується у веб-додатках, особливо з популярним стеком LAMP [25] (Linux, Apache, MySQL, PHP/Python/Perl). MySQL доступна у комерційній та безкоштовній відкритій версії (MySQL Community Edition).

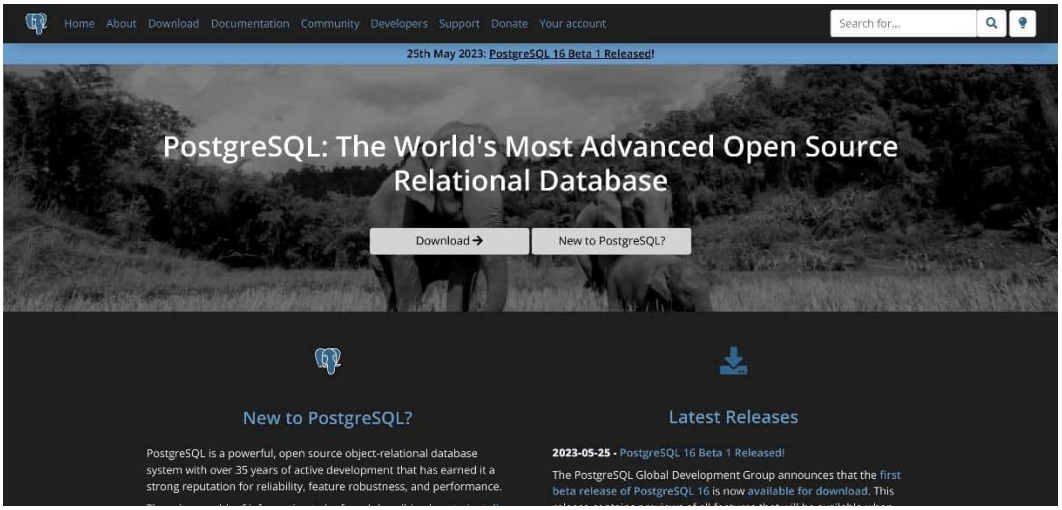

Рис. 2.7. Головна сторінка СУБД PostgreSQL

PostgreSQL є потужною об’єктно-реляційною СУБД з відкритим вихідним кодом (Головна сторінка представлена на Рис. 2.7). Вона підтримує широкий спектр функцій, включаючи реляційні, об’єктно-орієнтовані та географічні типи даних. PostgreSQL надає розширені можливості для складних запитів та операцій з базами даних. Вона підтримує мову SQL і включає можливості для розробки процедур на мовах програмування, таких як SQL, Python, Perl, Java та інші.

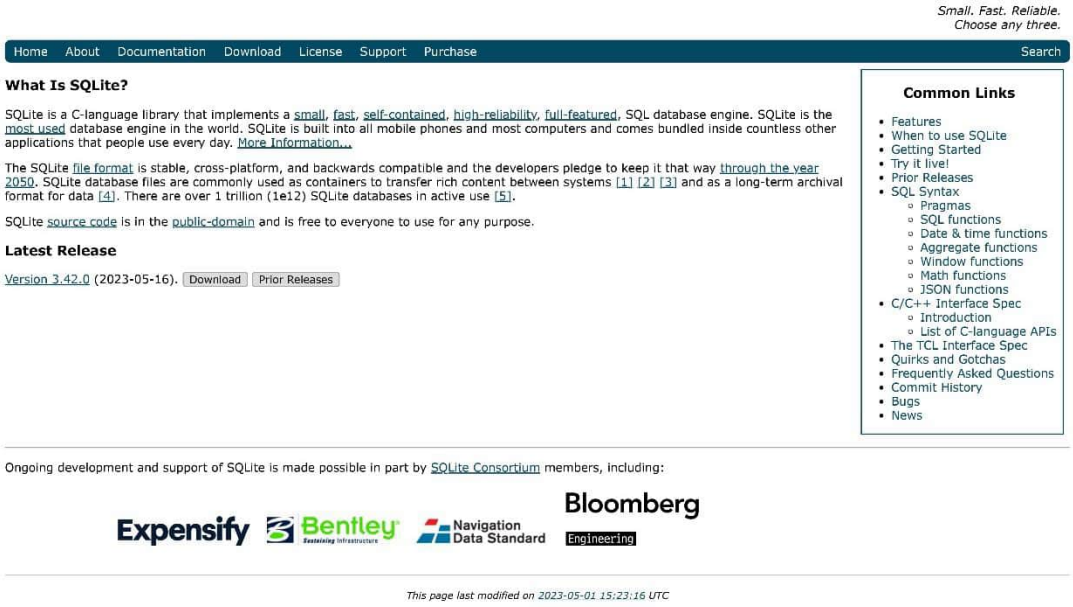

Рис. 2.8. Головна сторінка СУБД SQLite

SQLite є легкою вбудованою СУБД, яка не вимагає окремого сервера баз даних (Головна сторінка представлена на Рис. 2.8). Вона зберігає всю базу даних

в одному файлі, що робить її простою у використанні та переносі. SQLite підтримує більшість стандартних функцій SQL та може використовуватись в багатьох типах додатків, включаючи мобільні додатки та веб-сайти з низьким рівнем навантаження.

Кожна з цих СУБД має свої переваги та обмеження, і вибір між ними залежить від конкретних потреб проекту. MySQL зручно використовувати для веб-сайтів та додатків, PostgreSQL – для складних аналітичних завдань та проектів, де потрібна розширена функціональність, а SQLite – для вбудованих або простих додатків з низьким рівнем навантаження.

Проаналізувавши вище наведені дані, було зроблено висновок, що найдоцільніше для цього дипломного проєкту використовувати саме SQLite.

2.5. Фреймворк Bootstrap для розробки веб-інтерфейсів

Bootstrap є одним з найпопулярніших фреймворків для розробки веб-інтерфейсів [26]. Він надає набір готових компонентів, стилів і JavaScript-плагінів, що допомагають швидко та ефективно створювати сучасні та респонсивні веб-сторінки. Головна сторінка Bootstrap наведена на Рис. 2.9.

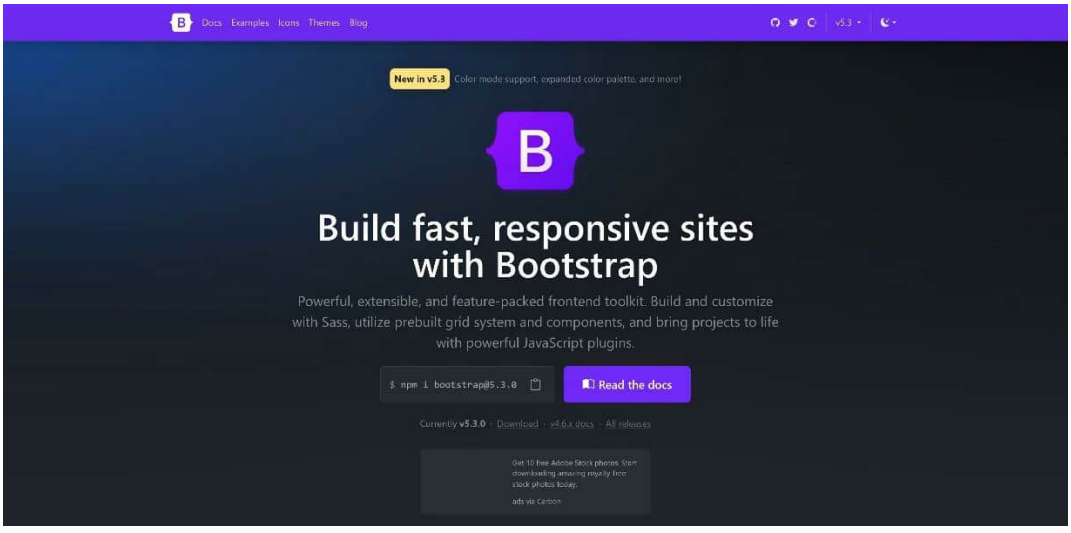

Рис. 2.9. Головна сторінка фреймворку Bootstrap

Bootstrap має велику кількість готових компонентів, таких як кнопки, форми, навігаційні панелі, каруселі, модальні вікна та інші. Використовуючи ці

компоненти, можна швидко побудувати функціональний та стильний інтерфейс для проєкту.

Bootstrap має чистий та зрозумілий код, що дозволяє легко розуміти та змінювати компоненти за потребою. Він також має докладну документацію та приклади, які допомагають швидко орієнтуватися у фреймворку та використовувати його ефективно.

Що не менш важливо, фреймворк забезпечує сумісність з різними веб-браузерами, включаючи останні версії Chrome, Firefox, Safari, Internet Explorer та інших популярних браузерів. Це дозволить проєкту працювати на широкому спектрі платформ та браузерів без проблем з відображенням.

2.6. Система управління версіями Git

Git є розподіленою системою управління версіями, яка дозволяє контролювати та відстежувати зміни у вихідному коді проекту [27]. Головна сторінка Git наведена на Рис. 2.10.

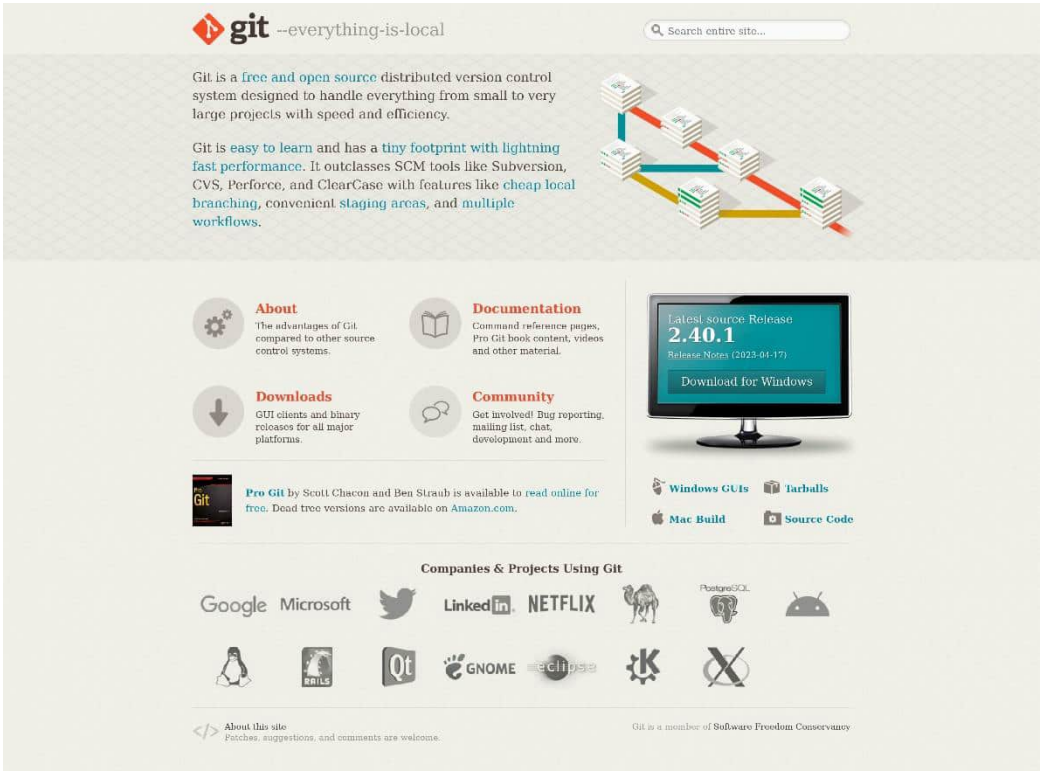

Рис. 2.10. Головна сторінка системи управління версіями Git

Git зберігає всі зміни, які були внесені у проєкт, у вигляді версій. Це означає, що можна легко переглядати, порівнювати та відновлювати попередні стани проекту. Якщо зроблена помилка або потрібно повернутися до попередньої версії коду, Git допоможе зробити це без проблем. Ця властивість Git графічно зображена на Рис. 2.11.

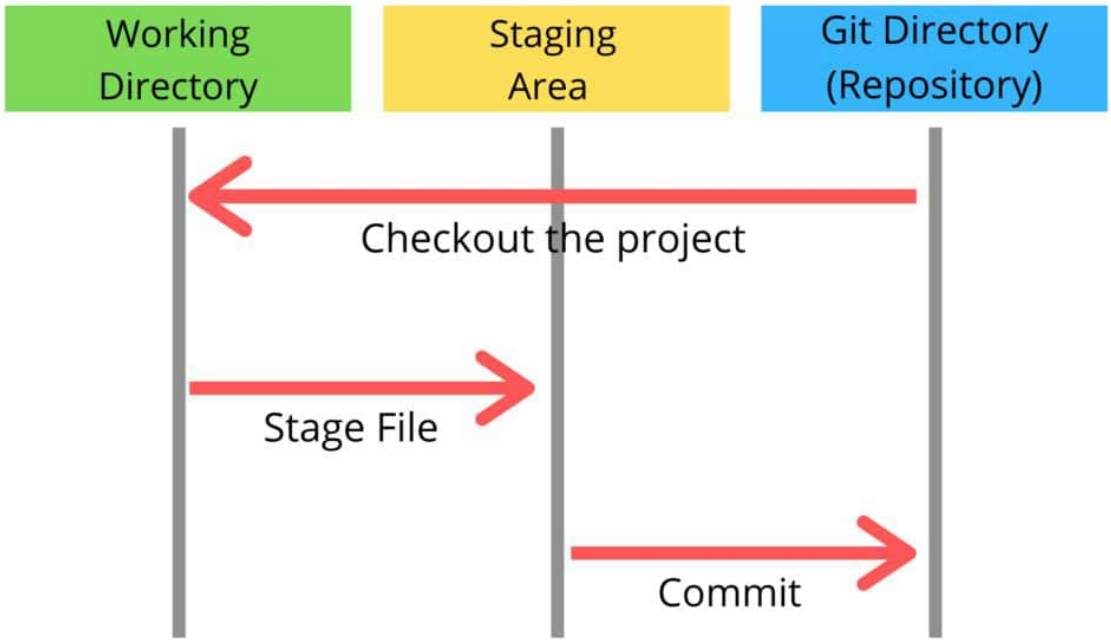

Рис. 2.11. Відображення роботи 3-ох рівневої архітектури Git

Git дозволяє розмістити репозиторій на веб-сервісі, такому як GitHub [28] або GitLab [29], що дозволяє зберігати проект в хмарі та мати доступ до нього з будь-якого місця.

Тому, використання Git у проєкті допоможе зберегти контроль над версіями коду, ефективно вести розробку та забезпечити безпеку даних.

2.7 Принципи використання API Chat-GPT

API Chat-GPT – це інтерфейс програмування застосунків, який надає доступ до моделі GPT для взаємодії з текстом (Головна сторінка сервісу наведена на Рис. 2.12). Завдяки цьому API можна використовувати потужні можливості мовного моделювання GPT для розв’язання різних завдань, таких як генерація тексту, відповіді на запитання та багато інших.

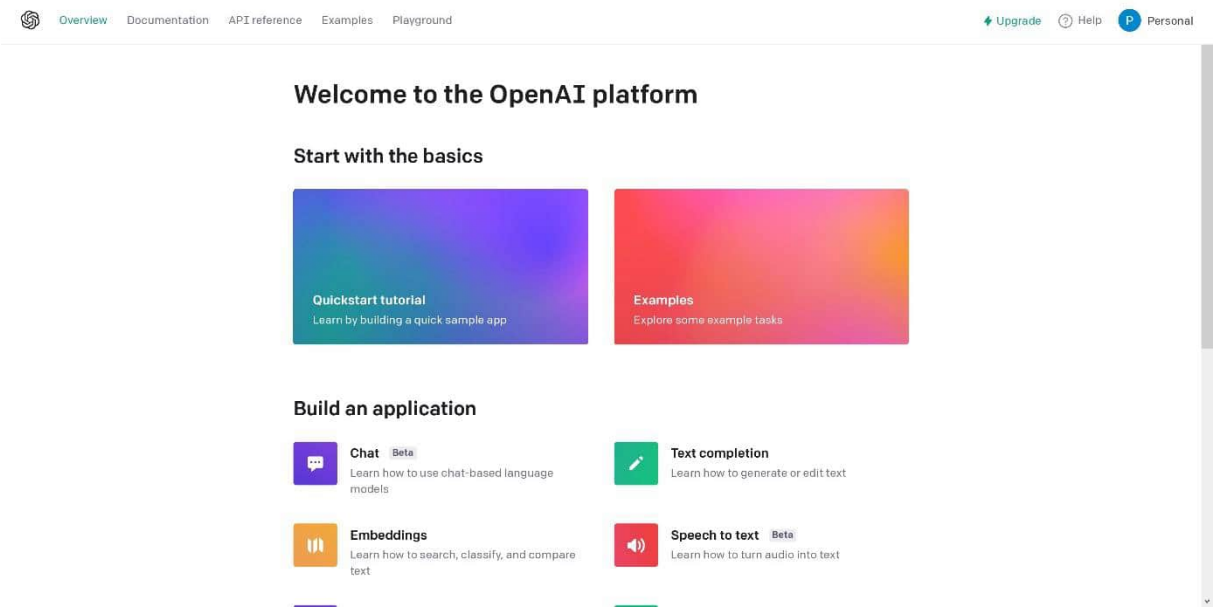

Рис. 2.12. Головна сторінка платформи OpenAI

Використання API Chat-GPT дозволяє додати до проекту потужні можливості мовного моделювання. Можна створити інтерактивного чат-бота, який відповідає на запитання користувачів, надає поради або здійснює інші завдання, пов’язані з обробкою тексту. Використання API дозволяє використовувати сучасну технологію мовного моделювання, яка постійно розвивається і оновлюється. Це забезпечує доступ до оновлень моделі і можливостей, що дозволяє проєкту залишатися актуальним у сфері обробки тексту.

Загалом, застосування API Chat-GPT може додати значну цінність до дипломного проєкту, надає йому більші можливості та покращує взаємодію з користувачем.

РОЗДІЛ 3. ОПИС РОЗРОБЛЕНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ГОТЕЛЮ

3.1 Огляд функціональних можливостей та інтерфейсу системи готелю

Інформаційна система готелю є веб-ресурсом, розробленим на основі мови програмування Python з використанням фреймворку Django. Клієнти-користувачі отримують доступ до функціоналу системи через веб-сторінку, яка відображається у веб-браузері.

Загалом, цей ресурс надає користувачам такі можливості:

Для користувачів (клієнтів):

- Перегляд розташування готелю на Google Maps;
- Перегляд номерів та їхніх детальних характеристик;
- Перегляд послуг готелю;
- Додавання коментарів до готельних номерів;
- Перегляд існуючих коментарів на сайті;
- Можливість задати питання, яке стосується готелю розумному помічнику;
- Перехід по контактних сторінках готелю.

Для персоналу:

- Засоби керування Django Admin [30].

Інформаційна система готелю має дві сторони:

Front-end – це клієнтська сторона (підсистема), яка забезпечує зручний доступ до функціоналу системи через веб-браузер для всіх користувачів, навіть з мінімальними навичками роботи з комп’ютером.

Back-end – це серверна сторона (підсистема), яка обробляє всі запити користувачів, виконує обробку форм, взаємодіє з базою даних, що приєднана до системи.

3.1.1 Клієнтська підсистема

Розглянемо детальніше можливості що надає клієнтська підсистема. Вона реалізована у вигляді сайту та має два основних шляхи: /admin/ та /webhotel/.

/admin/ (Рис. 3.1) являє собою автоматично згенеровану сторінку інтерфейсу адміністратора. У панелі адміністратора Django працівники можуть виконувати різноманітні дії для управління веб-додатком, як-от: управління моделями, управління користувачами та групами, налаштування прав доступу та редагування вмісту сайту.

Рис. 3.1. Загальний вигляд сторінки /admin/

Для забезпечення комфортного використання системи на різних пристроях та спрощення процесу верстки сторінок було використано адаптивний дизайн та функціонал front-end фреймворку Bootstrap (Версія 5.2.3).

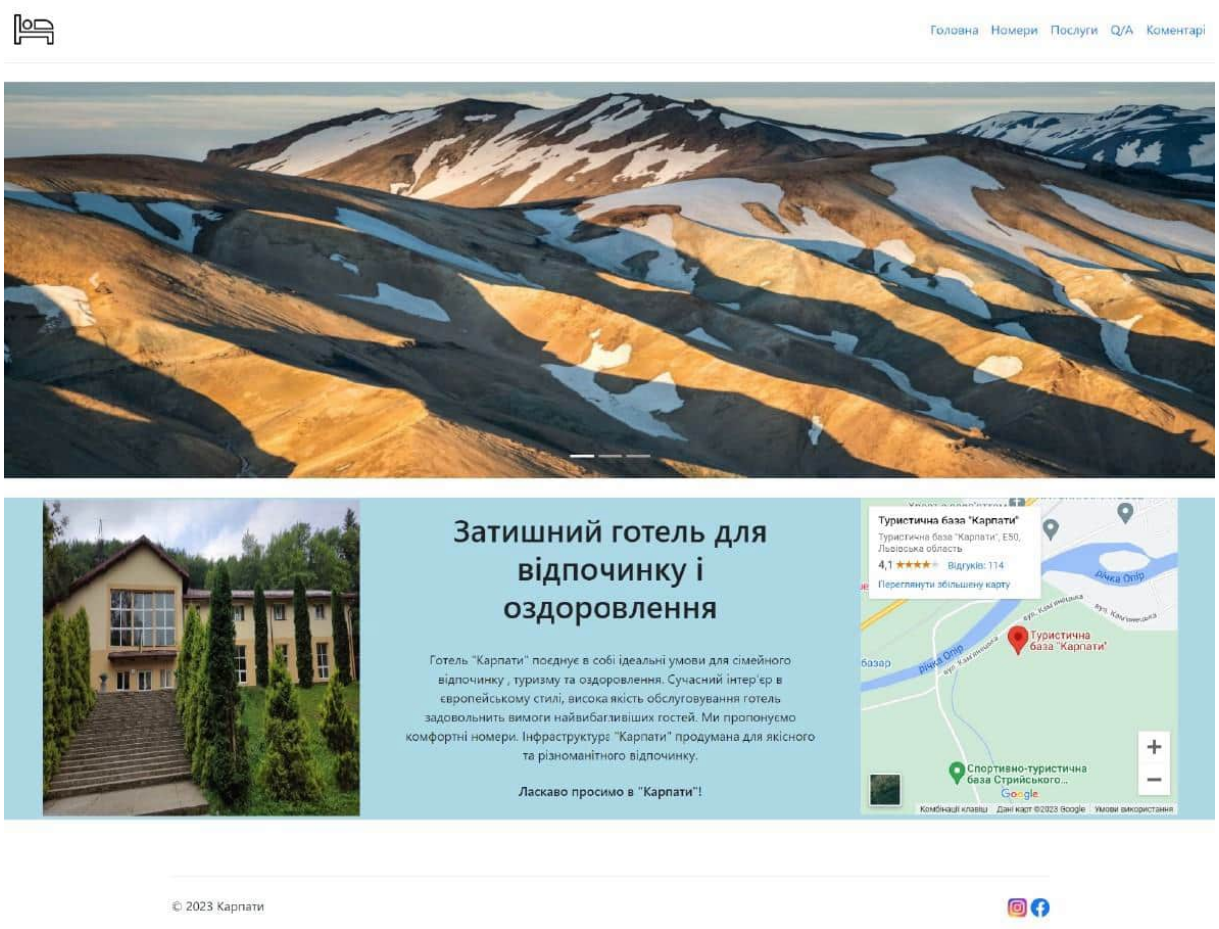

Рис. 3.2. Загальний вигляд головної сторінки сайту

Основні елементи головної сторінки сайту (Рис. 3.2):

- Навігаційна панель. Містить пункти навігації, доступні користувачу;
- Карусель. Містить фото, які рухаються зліва направо;
- Таблиця. Поділяється на 3 елементи (зліва направо):
    - Фото готелю;
    - Короткий опис готелю;
    - Розташування готелю.
- Футер. Містить 2 елементи (зліва направо):
    - Відмітка “авторські права”;
    - Контактні сторінки готелю.

Для компонування сторінки використовується сітка на основі системи Bootstrap, яка базується на принципах flexbox [31]. Вміст деяких елементів сторінок сайту досягається завдяки внутрішній роботі шаблонів Django. Приклад такої адаптації наведено на Рис 3.3.

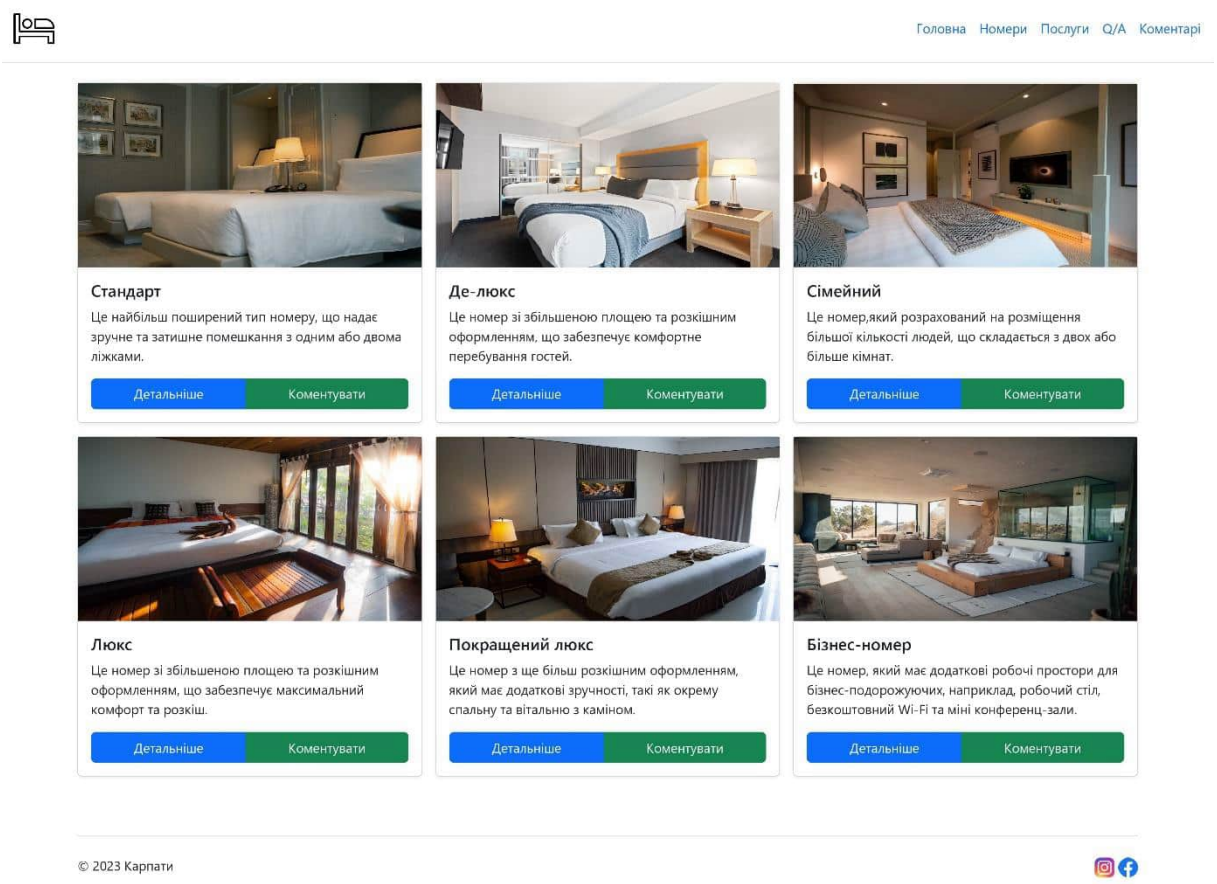

Рис. 3.3. Загальний вигляд сторінки з наявними номерами готелю на сайті

Основні елементи сторінки, які можна побачити на Рис. 3.3, містять 6 блоків таблиці шаблону Bootstrap, у яких міститься:

- Основне фото;
- Тип номеру;
- Короткий опис номеру;
- Кнопка “Детальніше”. Містить посилання на сторінку, де відображається детальна інформація про тип номеру.
- Кнопка “Коментувати”. Містить посилання на сторінку, де користувач може коментувати.

Відповідно до шаблону сторінки з наявними номерами готелю, загалом є 6 типів номерів: “Стандарт”, “Де-люкс”, “Сімейний”, “Люкс”, “Покращений люкс” та “Бізнес-номер”. Усі містять однакову структуру по посиланні кнопки “Детальніше”, проте різний вміст. Тому, для прикладу, розглянемо сторінку, де можна поглянути детальний опис типу номеру “Стандарт” (Рис. 3.4).

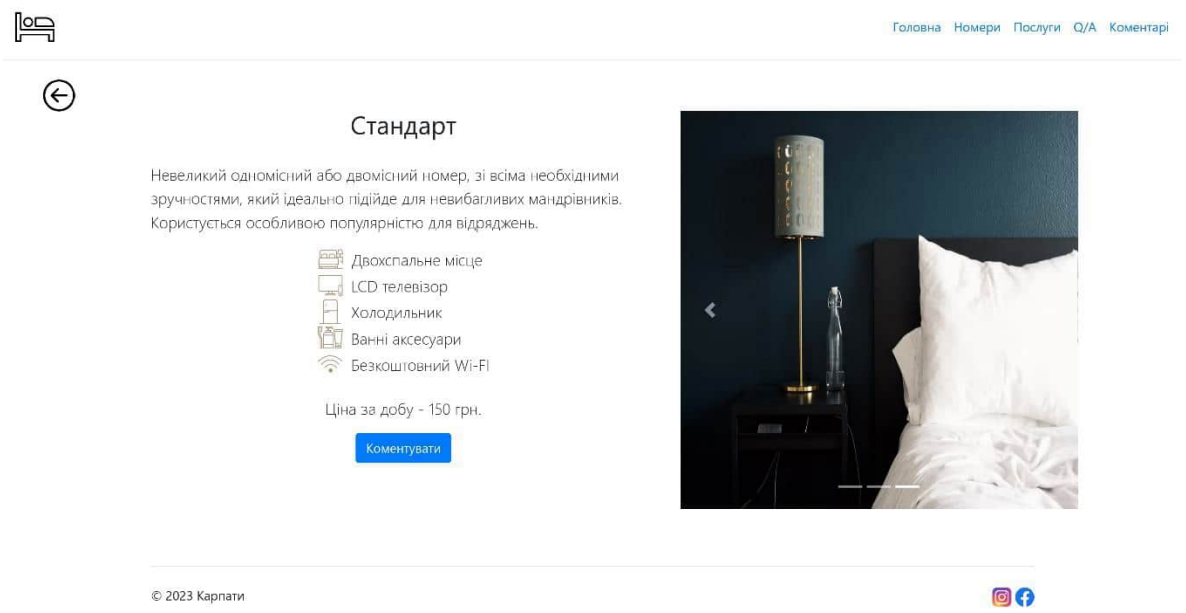

Рис. 3.4. Загальний вигляд сторінки детального опису типу номеру “Стандарт”

На сторінці сайту, де використовується API Chat-GPT (Рис. 3.5), користувачі можуть насолоджуватись функціоналом інтелектуального чат-бота, який забезпечує автоматичну обробку та відповідь на їх запитання. Це API дає можливість інтегрувати потужну мовну модель GPT у веб-додатки та взаємодіяти з нею за допомогою програмного інтерфейсу.

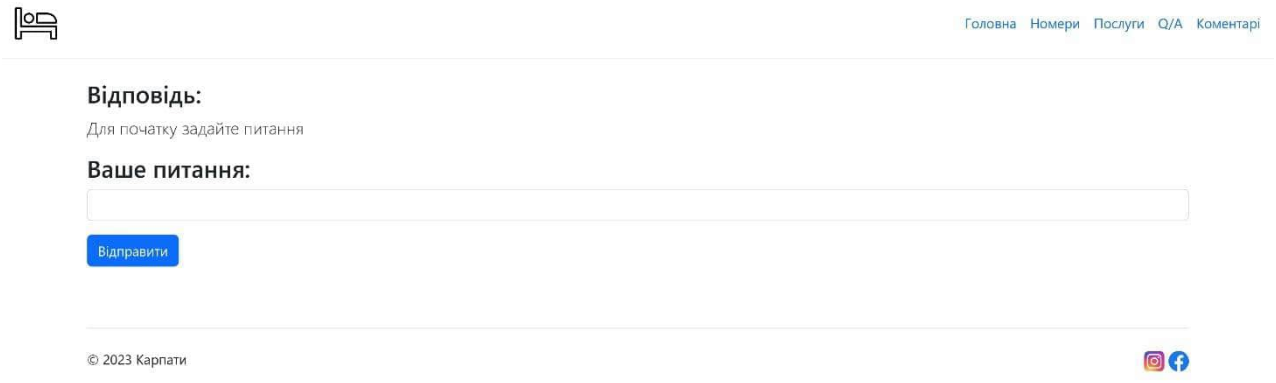

Рис. 3.5. Загальний вигляд сторінки, на якій застосовується API Chat-GPT

На цій сторінці користувач може ввести свої запитання або повідомлення в текстове поле, що наведено на Рис. 3.6 і надіслати їх на обробку через API Chat-GPT.

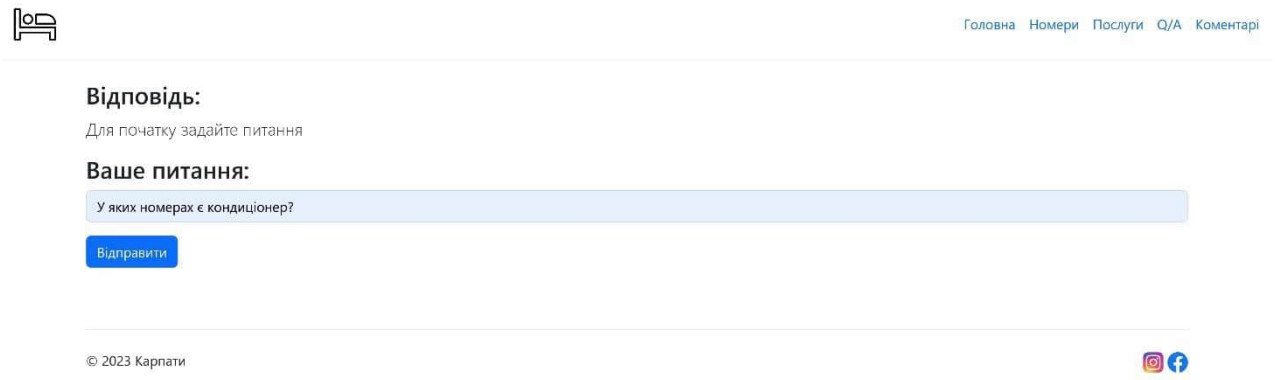

Рис. 3.6. Загальний вигляд сторінки, на якій застосовується API Chat-GPT при введенні питання користувачем

Після цього система виконає аналіз тексту та згенерує відповідь на основі свого навчання та попередньої обробки. Згенерована відповідь буде відображена на сторінці (Див. Рис. 3.7), дозволяючи користувачеві отримати необхідну інформацію.

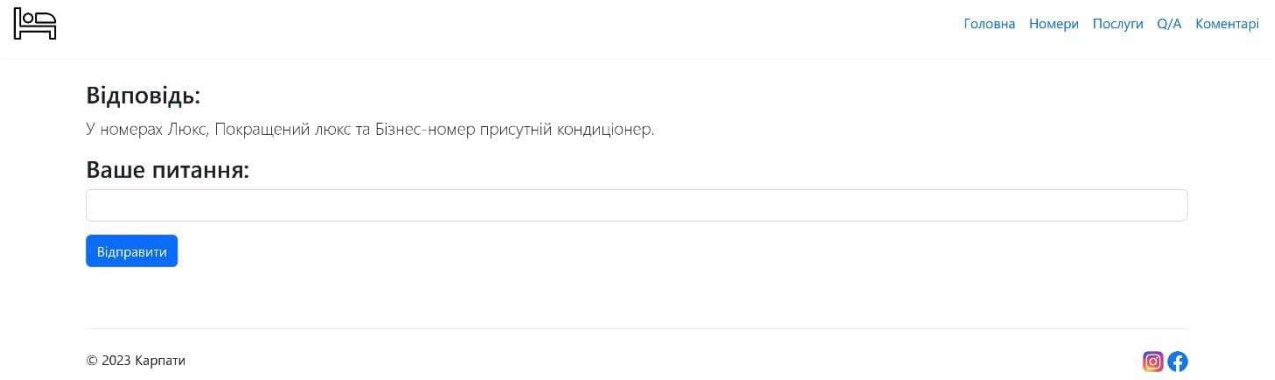

Рис. 3.7. Загальний вигляд сторінки, на якій застосовується API Chat-GPT при відповіді на запитання, що попередньо ввів користувач

На сайті також є можливість взаємодіяти з номерами та обмінюватись думками з іншими користувачами за допомогою зручного та зрозумілого інтерфейсу (Рис. 3.8).

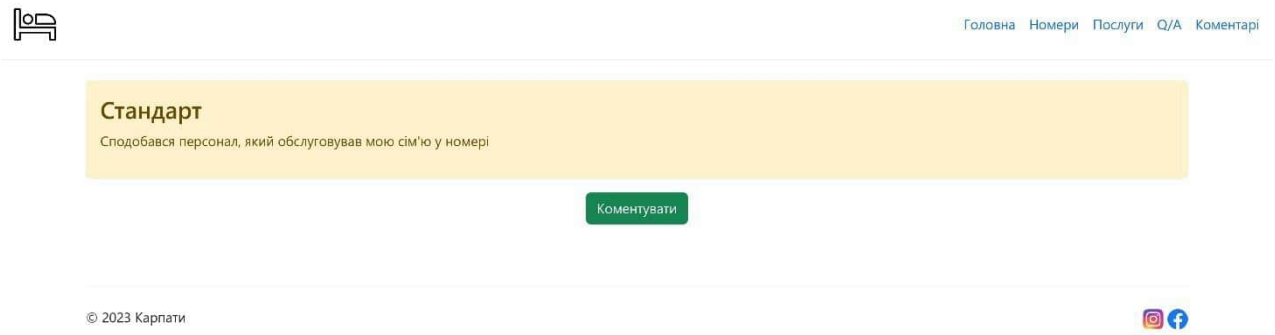

Рис. 3.8. Загальний вигляд сторінки, у якій користувач може коментувати номери та переглядати попередні коментарі

Сторінка містить наступні компоненти:

- Блок коментарів;
- Форма коментування.

Користувач може бачити блок коментарів, де відображаються попередні коментарі інших користувачів. Кожен коментар назву та основний текст коментаря.

Під блоком коментарів розташована кнопка “Коментувати” з посиланням на форму, де користувач може ввести свій коментар (Рис. 3.9).

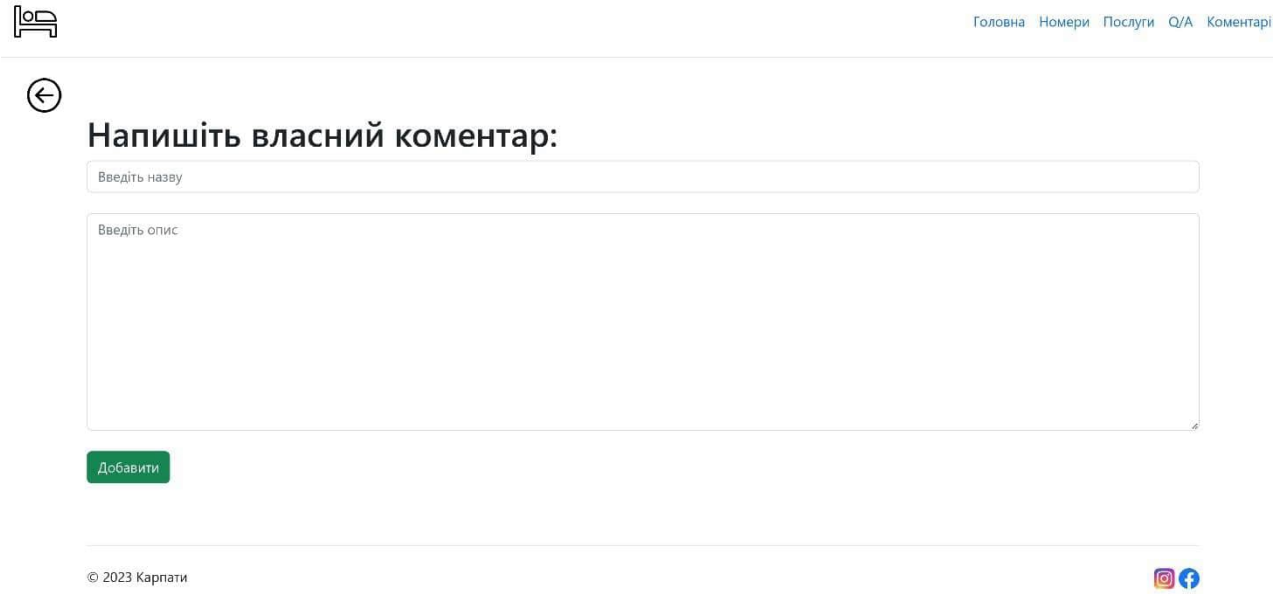

Рис. 3.9. Загальний вигляд сторінки для добавлення коментарів на веб-сайт

Сторінка для добавлення коментарів дозволяє користувачам висловлювати свою думку та ділитися своїми враженнями. Сторінка містить наступні компоненти:

- Заголовок: На верхній частині сторінки розташований заголовок “Власний коментар”;
- Форма коментування: Головною частиною сторінки є форма, де користувач може ввести свій коментар. Форма містить 2 текстових поля для написання назви коментаря та основного опису;
- Кнопка надіслати: Знизу від форми коментування знаходться кнопка “Добавити”, яку користувач може натиснути, щоб надіслати свій коментар.

3.1.2 Серверна підсистема

Тепер розглянемо можливості, що надає серверна підсистема. Ця система реалізована у вигляді Django додатку і включає в себе всі стандартні можливості фреймворку.

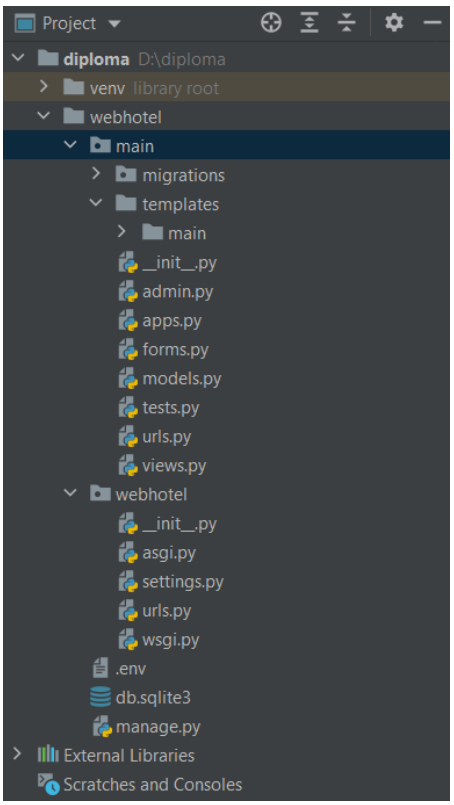

Рис. 3.10. Структура розробленої інформаційної системи готелю

Основні аспекти структури (Рис. 3.10) містять наступні елементи:

- Папка “migrations”. У папці містяться міграції, які проводяться під час змін у БД;
- Папка “templates”. У папці містяться html-шаблони для графічного відображення функціоналу системи;
- Папка “main”. Містить функціональне наповнення, розглянемо основні з них:
    - “forms.py”. Клас, який дозволяє налаштовувати моделі;
    - “models.py”. Файл, який створює моделі;
    - “urls.py”. Файл для виводу html-шаблонів;
    - “views.py”. Файл, у якому задаються параметри для виводу html-шаблонів.
- Папка “webhotel”. Ця папка містить конфігуруючі файли системи:
    - “asgi.py”. Клієнт-серверний протокол взаємодії веб-сервера та програми;
    - “settings.py”. Файл містить параметри системи;
    - “urls.py”. Файл для виводу html-шаблонів з папки “main”;
    - “wsgi.py”. Стандарт взаємодії програми Python.
- Файл “env”. Містить API ключ Chat GPT;
- Файл “db.sqlite3”. База даних проєкту;
- Файл “manage.py”. Виконувальний файл.

Система може працювати як серверний додаток і виконувати різні операції, окрім запуску сервера, можлива також робота в командному режимі (shell), генерація та застосування міграцій бази даних, а також виконання додаткових розроблених команд.

Функціональні можливості сервера включають наступне:

- Синхронний зв’язок з користувачами, що дозволяє взаємодіяти з ними в реальному часі.

- Взаємодія з базою даних, включаючи отримання, обробку, збереження, надсилання користувачам та видалення даних. В якості бази даних використовується Sqlite.
- Обробка даних з форм, що дозволяє користувачам взаємодіяти з системою шляхом введення та відправки інформації через форми.
- В системі інтегровано API Chat-GPT, що дозволяє користувачам задавати запитання та отримувати автоматичні відповіді на їх запитання, пов’язані з готелем, послугами чи будь-якою іншою інформацією, доступною в системі.

Ці функції сприяють роботі системи, забезпечують взаємодію з користувачами, збереження та обробку даних, а також контроль доступу до різних функціональних можливостей.

Розроблений додаток є інтерактивним, і більшість взаємодії з користувачами відбувається за синхронною моделлю (Рис. 3.11), де дані передаються між клієнтом і сервером в реальному часі.

```
[04/Jun/2023 15:10:05] "GET / HTTP/1.1" 200 6058
[04/Jun/2023 15:10:07] "GET /services HTTP/1.1" 200 13332
[04/Jun/2023 15:10:08] "GET /qa HTTP/1.1" 200 2715
[04/Jun/2023 15:10:09] "GET /create_comment HTTP/1.1" 200 2646
[04/Jun/2023 15:10:14] "GET /about HTTP/1.1" 200 8964
[04/Jun/2023 15:10:15] "GET / HTTP/1.1" 200 6058
[04/Jun/2023 15:10:20] "GET /about HTTP/1.1" 200 8964
[04/Jun/2023 15:29:32] "GET / HTTP/1.1" 200 6058
[04/Jun/2023 15:29:33] "GET /qa HTTP/1.1" 200 2715
[04/Jun/2023 15:29:35] "GET /create_comment HTTP/1.1" 200 2646
[04/Jun/2023 15:36:23] "GET /create HTTP/1.1" 200 3035
[04/Jun/2023 15:36:26] "GET /create_comment HTTP/1.1" 200 2646
[04/Jun/2023 15:44:01] "GET /create HTTP/1.1" 200 3035
```

Рис. 3.11. Графічне представлення з консолі синхронних запитів у системі

На сайті використовуються моделі для визначення структури та поведінки даних. Вони представляють об’єкти, які відповідають таблицям бази даних і

забезпечують зручний спосіб взаємодії з даними. Використання моделей на сайті спрощує роботу з даними і дозволяє ефективно керувати інформацією. Вони забезпечують структурований підхід до управління даними, дозволяючи легко взаємодіяти з базою даних і виконувати необхідні операції.

У веб-системі також використовується API Chat-GPT для надання можливості взаємодії зі штучним інтелектом через чат. Це дозволяє користувачам спілкуватися з системою, задавати запитання і отримувати відповіді в реальному часі.

Щоб реалізувати взаємодію сайту з API Chat-GPT для початку потрібно зареєструватись на офіційному сайті OpenAI та перейти до створення API ключа (Рис. 3.12).

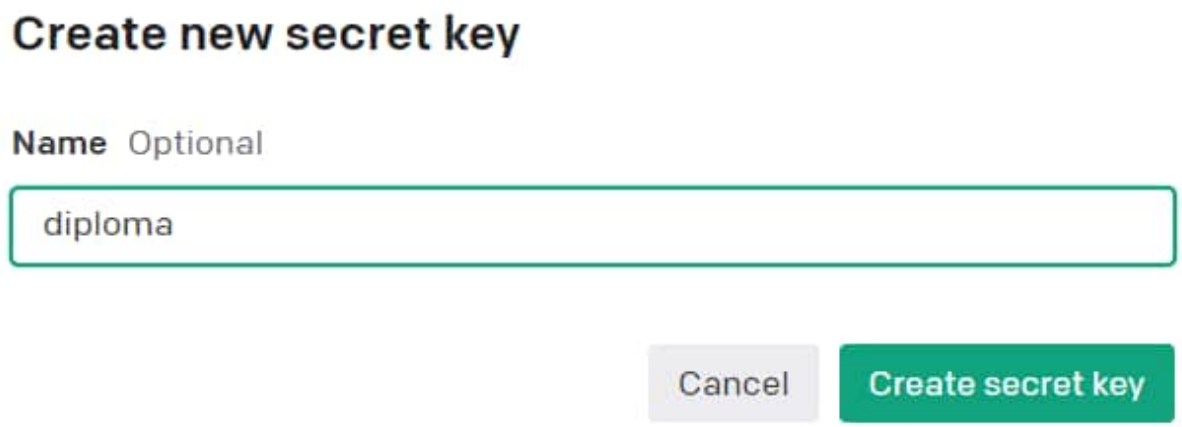

Рис. 3.12. Вікно створення API ключа Chat-GPT від OpenAI

Після створення ключа, згідно з документацією, яку надає сайт OpenAI, у проект потрібно додати файл “.env” і помістити туди змінну “OPENAI_KEY” з вмістом згенерованого ключа (Рис. 3.13).

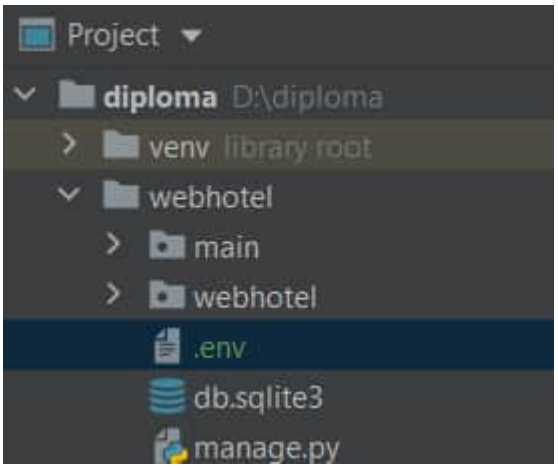

Рис. 3.13. Відображення файлу “.env” у проєкті

Розробивши функціонал відображення у html-шаблоні та файлі “views.py”, заповнюємо форму для відправлення звернення до Chat-GPT (Рис. 3.6) та отримуємо такий результат у терміналі, що зображений на Рис. 3.14.

```
[13/May/2023 16:07:38] "GET / HTTP/1.1" 200 5964
[13/May/2023 16:07:50] "GET /qa HTTP/1.1" 200 2492
{
  "choices": [
    {
      "finish_reason": "stop",
      "index": 0,
      "logprobs": null,
      "text": "\n\nHello! It's great to meet you!"
    }
  ],
  "created": 1683983342,
  "id": "cmpl-7FjHa4bKQJwqVU1FWyxTJUwPD8cIq",
  "model": "text-davinci-003",
  "object": "text_completion",
  "usage": {
    "completion_tokens": 11,
    "prompt_tokens": 3,
    "total_tokens": 14
  }
}
```

Рис. 3.14. Відображення методу “GET” від Chat-GPT

3.2 Шляхи подальшого потенційного розвитку системи

Для подальшого потенційного розвитку системи є кілька шляхів, які можуть сприяти поліпшенню її функціональності та користувацького досвіду. Перш за все, можна розширити функціональність системи шляхом додавання нових функцій і можливостей, що відповідають зростаючим потребам користувачів.

Далі, покращення інтерфейсу користувача може значно покращити взаємодію з системою. Дослідження сучасних технологій, використання анімацій, покращена навігація та оптимальне розміщення контенту допоможуть забезпечити зручний та привабливий дизайн.

І, нарешті, можна і потрібно вдосконалювати технологію AI, оскільки даний метод ведення бізнесу реально має потенціал. Використання методів штучного інтелекту, таких як машинне навчання, глибоке навчання та обробка природної мови, може привести до значних переваг і ефективності в різних сферах. Наприклад, система може використовувати алгоритми машинного навчання для автоматичної обробки великого обсягу даних про клієнтів, що допоможе виявити патерни та тренди і зробити більш точні прогнози та рекомендації.

Отже, можна підсумувати, що пріоритет для подальшого розвитку системи слід надавати технології штучного інтелекту, адже вона є потужним інструментом для покращення бізнесу та оптимізації інформаційних систем. Вдосконалення технології AI може привести до збільшення продуктивності, ефективності та конкурентоспроможності системи, відкриваючи нові можливості для розвитку та розширення бізнесу.

ВИСНОВКИ

Застосування штучного інтелекту у створенні веб-сайту дало змогу покращити обслуговування клієнтів та збільшити ефективність бізнес-процесів. Створення “розумного консультанта” на основі API Chat-GPT, уможливлює аналіз, класифікацію та відповіді на запити користувачів в режимі реального часу, забезпечуючи швидке та ефективне обслуговування без участі людей. Завдяки автоматизованому чат-боту, клієнти можуть отримувати відповіді на свої запитання та підтримку у будь-який час, навіть поза робочими годинами готелю. Це покращує задоволеність відвідувачів та сприяє збереженню часу і зусиль для них.

Завдяки ефективно спроектованій базі даних можливо швидко отримувати доступ до коментарів та відгуків клієнтів. Це забезпечує оперативну реакцію на запити користувачів та підвищує якість обслуговування. Моделі баз даних дозволяють структуровано зберігати коментарі гостей номерів. Обсяг інформації про враження відвідувачів допомагає покращити взаємодію з потенційними клієнтами.

Загалом, розробка цієї інформаційної системи готелю є кроком у напрямку цифрової трансформації готельного господарства. Вона не лише сприяє автоматизації та оптимізації бізнес-процесів, але й покращує взаємодію з клієнтами, забезпечуючи їм зручний доступ до інформації та послуг. З врахуванням потенціалу та можливостей API Chat-GPT, ця технологія може стати конкурентоспроможним інструментом у сфері готельного бізнесу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Яковенко А. В. Основи програмування. Python. Частина 1. КПІ ім. Ігоря Сікорського. Київ, 2018. С. 41-105.
2. OpenAI /ChatGPT – як використовувати його з Python. URL: https://blog.devgenius.io/chatgpt-how-to-use-it-with-python-5d729ac34c0d.
3. Сталіров В. Gdpr compliance: що це та кому потрібно виконати умови регламенту про захист персональних даних. 2022. URL: https://stalirov.lawyer/uk/posts/gdpr-dlya-ukrayins-kih-it-kompanij.
4. О'Феллон М., Резерфорд Д. Управління та експлуатація готелю, 5-е видання. 2010. 512 с.
5. МакГаррі Вулф М. Цифровий маркетинг у готельній індустрії: Посібник із розуміння інструментів і методів онлайн-маркетингу та соціальних медіа. Каліфорнійський політехнічний державний університет. 2012.
6. Шоул Д. Першокласний сервіс як конкурентна перевага. Альпіна Паблішер. 2008. 344 с.
7. Маккеллі Р. Управління маркетинговими каналами. Praeger. 1996. 262 с.
8. МакТір М. Розмовний ШІ. Системи діалогу, розмовні агенти та чат-боти. Видавництво Morgan & Claypool. 2020. 251 с.
9. Шпет К. SWOT-аналіз. Важливий інструмент для розробки бізнес-стратегій. 50Minutes.com. 2023. 29 с.
10. Результати опитування Stack Overflow 2019. URL: https://codeguida.com/post/1773.
11. Вінсент В. Django для API. Створюйте веб-інтерфейси API за допомогою Python і Django. Amazon Digital Services LLC – Kdp. 2020. 210 с.
12. Грінберг М. Розробка веб-застосунків з використанням Flask мовою Python. ЛітРес. 2022.
13. Тхань Н. Веб-розробка Flask. Розробка веб-додатків на Python. Тхань Н. 2019. С. 25-32.

14. Система міграції у Django. URL: https://highload.today/uk/blogs/vse-ne-tak-prosto-yak-zdayetsya-rozbirayemo-skladni-vipadki-migratsiyi-baz-danih-v-django/.
15. Маєрс Д., Коупленд Р. Основна SQLAlchemy. Відображення Python у базах даних. O'Reilly Media. 2015. 208 с.
16. Офіційний сайт Django. URL: https://www.djangoproject.com.
17. Білоус А., Солодуха В. Основи кібербезпеки. Стандарти, концепції, методи та засоби забезпечення. Літрес. 2021.
18. Міжнародна наукова група. Сучасні методи застосування наукових теорій. Primedia eLaunch LLC. 2023. С. 457-458.
19. Портела М., Крус-Кунья М. Довідник з досліджень цифрових злочинів, безпеки кіберпростору та забезпечення інформації. IGI Global. 2014. С. 144-145.
20. X-Frame-Options – Як боротися з Clickjacking. URL: https://www.keycdn.com/blog/x-frame-options.
21. Лоу С., Елдман Д., Осуолт М. Автоматизація програмованих мереж. Літрес. 2022. С. 215-217.
22. Офіційний сайт MySQL. URL: https://www.mysql.com.
23. Офіційний сайт PostgreSQL. URL: https://www.postgresql.org.
24. Офіційний сайт SQLite. URL: https://sqlite.org/index.html.
25. Вікіпедія. Стек LAMP. URL: https://uk.wikipedia.org/wiki/LAMP.
26. Летт Д. Короткий старт Bootstrap 4: посібник для початківців зі створення адаптивних макетів за допомогою Bootstrap 4. Bootstrap Creative. 2018. 200 с.
27. Маккуейд М. Git на практиці. Manning. 2014. 272 с.
28. Гуталс С. GitHub для чайників. Wiley. 352 с.

29. О’Грейді А. Короткий посібник GitLab. Перейдіть на GitLab, щоб отримати всі свої рішення для керування репозиторіями. Packt Publishing. 2018. 180 с.

30. Офіційна документація Django. Сайт адміністратора Django. URL: https://docs.djangoproject.com/en/4.2/ref/contrib/admin/.

31. Вейль Е. Flexbox в CSS. O’Reilly Media. 2017. 153 с.