

КВАЛІФІКАЦІЙНА РОБОТА

Група ІПЗс 20-1

Батіг В.М

2024

ЗВО УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА

Факультет суспільних та прикладних наук

Кафедра інформаційних технологій

на правах рукопису

Батіг Вікторія Миколаївна

УДК 004.4

**Розробка веб-сайту для організації рецептів з інтеграцією відео з
YouTube**

Спеціальність 121 – «Інженерія програмного забезпечення»

Кваліфікаційна робота на здобуття кваліфікації бакалавр

Нормоконтроль

_____ Сτισло О.В.

(підпис, дата, розшифрування підпису)

Студент

_____ Батіг В.М

(підпис, дата, розшифрування підпису)

Допускається до захисту

Завідувач кафедри

_____ к.т.н., доц. Ващишак С.П.

(підпис, дата, розшифрування підпису)

Керівник роботи

асистент каф.ІТ

_____ Шкатуляк В.В

(підпис, дата, розшифрування підпису)

Івано-Франківськ – 2024

ЗВО УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА
Факультет суспільних та прикладних наук
Кафедра інформаційних технологій

Освітній ступінь: «бакалавр»

Спеціальність: 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

« ____ » _____ 2024 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Батіг Вікторія Миколаївна

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:

Розробка веб-сайту для організації рецептів з інтеграцією відео з YouTube

керівник роботи:

Шкатуляк Василь Васильович, асистент каф.ІТ

затверджена наказом вищого навчального закладу від « 12 » березня 2024 року

№ 19/1

2. Термін подання студентом роботи 05.06.2024

3. Вихідні дані роботи: Мова програмування Python, веб-фреймворк

Django, ключ API YouTube

4. Зміст кваліфікаційної роботи (перелік питань, які потрібно розробити)

1. Опис веб-ресурсів аналогів

2. Обґрунтування вибору технологій

3. Реалізація продукту

5. Дата видачі завдання 14.03.2024

КОНСУЛЬТАНТИ РОЗДІЛІВ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Розділ	Консультант (прізвище, ініціали та посада)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Загальні теоретичні відомості	28.03.2024	Виконано
2.	Обґрунтування вибору технологій	02.04.2024	Виконано
3.	Розробка структури продукту	22.04.2024	Виконано
4.	Розробка веб-ресурсу	03.05.2024	Виконано
5.	Оформлення графічного матеріалу	15.05.2024	Виконано
6.	Підготовка до захисту	24.05.2024	Виконано

Студент

(підпис)

Батіг В.М

(прізвище та ініціали)

Керівник роботи

(підпис)

Шкатуляк В.В

(прізвище та ініціали)

Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Сторінка	Опис графічного матеріалу	Сторінка	Опис графічного матеріалу
14	Огляд існуючого ресурсу YouTube	36	Зображення профілю користувача
15	Огляд існуючого ресурсу Allrecipes	38	Зображення функції додавання посилання на соціальні мережі
16	Приклад існуючого ресурсу Cookrade/ua/home	41	Приклад пошуку через YouTube API
17	Приклад аналогового ресурсу Tasty	43	“Нотатки”
24	Структура бази даних “PostgreSQL”	46	Форма редагування рецепту
24	Структура бази даних	49	Форма додавання рецепту
32	Зображення форми реєстрації користувача	49	Приклад роботи поля “Додавання рецепту”

АНОТАЦІЯ

У кваліфікаційній роботі розглядається розробка веб-сайту "Організатор рецептів з інтеграцією YouTube", що має на меті поліпшити досвід користувачів у кулінарії через інтеграцію навчальних відеоматеріалів безпосередньо з YouTube. Програма дозволяє користувачам легко знаходити, зберігати та категоризувати рецепти, а також переглядати пов'язані з ними навчальні відео, що робить процес навчання готуванню більш інтерактивним та доступним.

В першому розділі було досліджено вплив сучасних технологій на кулінарну сферу, а також був проведений аналіз вже існуючих ресурсів та виявлено проблему організації і ефективного використання кулінарного контенту через його велику кількість.

У другому розділі обрано технології для розробки веб-ресурсу "Організатор рецептів з інтеграцією YouTube". Python та Django обрані за швидкість розробки, масштабованість і підтримку спільноти. PostgreSQL забезпечує надійне управління даними. Bootstrap використовується для створення адаптивного інтерфейсу, доступного на різних пристроях. Інтеграція з YouTube API надає доступ до великої бази відео. Ці технології забезпечують ефективність, безпеку та зручність для користувачів, відповідаючи вимогам.

У третьому розділі розглянуто розробку веб-ресурсу "Організатор рецептів з інтеграцією YouTube". Описано процес створення форм реєстрації та входу, інтеграцію з YouTube API та систему нотаток для зручності користувачів. Профіль користувача став центральним місцем для персональної інформації та обміну рецептами, що заохочує активну взаємодію.

КЛЮЧОВІ СЛОВА: API, РОЗРОБКА, ОРГАНІЗАТОР, РЕЦЕПТИ, ІНТЕГРАЦІЯ, YOUTUBE.

SUMMARY

The qualification work addresses the development of the website "Recipe Organizer with YouTube Integration," aimed at enhancing the user experience in cooking by integrating instructional videos directly from YouTube. The program allows users to easily find, save, and categorize recipes, as well as view related instructional videos, making the cooking learning process more interactive and accessible.

The first chapter examines the impact of modern technologies on the culinary field and analyzes existing resources, identifying the problem of organizing and efficiently using culinary content due to its large volume.

In the second chapter, the technologies for developing the "Recipe Organizer with YouTube Integration" web resource were chosen. Python and Django were selected for their development speed, scalability, and community support. PostgreSQL ensures reliable data management. Bootstrap is used to create a responsive interface accessible on various devices. Integration with the YouTube API provides access to a vast video library. These technologies ensure efficiency, security, and convenience for users, meeting requirements.

The third chapter discusses the development of the "Recipe Organizer with YouTube Integration" web resource. The process of creating registration and login forms, YouTube API integration, and a note-taking system for user convenience is described. The user profile became a central place for personal information and recipe sharing, encouraging active interaction.

KEY WORDS: API, DEVELOPMENT, ORGANIZER, RECIPES, INTEGRATION, YOUTUBE.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	9
ВСТУП.....	10
РОЗДІЛ 1. АНАЛІЗ ПОТРЕБ ТА КОНКУРЕНТІВ.....	13
1.1 Виявлення потреби до організованого засобу з кулінарними рецептами.	13
1.2 Аналіз існуючих веб-ресурсів з кулінарії.....	14
1.3 Постановка задачі.....	17
Висновок до розділу 1.....	18
РОЗДІЛ 2. ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ РОЗРОБКИ.....	19
2.1 Обґрунтування вибору розробки організатора рецептів з інтеграцією YouTube.....	19
2.2 Обґрунтування вибору технологій та інструментів для розробки веб-ресурсу.....	21
2.3 Обґрунтування вибору мови Python.....	21
2.4 Обґрунтування вибору бази даних PostgreSQL.....	22
2.5 Структура БД.....	23
2.6 Обґрунтування вибору фреймворку Django.....	26
2.7 Обґрунтування вибору фреймворку Bootstrap.....	27
2.8 Використання API YouTube.....	28
2.9 Use Case Diagram проєкту.....	28
Висновок до розділу 2.....	30
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБ-РЕСУРСУ.....	32
3.1 Реалізація форми реєстрації та входу.....	32
3.3 Реалізація підключення YouTube API.....	39
3.4 Реалізація функціоналу “Нотаток”.....	41

	8
3.6 Створення форми додавання рецептів.....	47
Висновки до розділу 3.....	50
ВИСНОВКИ.....	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53
ДОДАТКИ.....	55
Додаток А.....	55

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

API	–	Application Programing Interface
СУБД	–	система управління базами даних
OAuth	–	Open Authorization
ORM	–	Object-Relational Mapping

ВСТУП

Актуальність теми. У цифрову епоху, коли інформація швидко рухається через онлайн-платформи, відео-контент займає центральне місце в житті користувачів Інтернету. YouTube, як один з найпопулярніших сервісів відеохостингу, є важливим джерелом інформації для мільйонів людей по всьому світу. У такому контексті, поєднання YouTube та організації рецептів стає особливо актуальною. Харчування є невід'ємною частиною нашого повсякденного життя, але часом вибір і приготування страв може бути викликом. Інформаційний потік у сучасному світі може перенаситити нас, і знайти якісний та доступний контент стає складніше. Саме тут на сцену виходить ідея "Організатора рецептів з інтеграцією YouTube". Ця робота ставить перед собою завдання створення зручного та ефективного інструменту, який би дозволяв користувачам з легкістю знаходити, організовувати та використовувати рецепти, інтегруючи в них відео-контент з YouTube. Розробка такого інструменту відкриває широкі перспективи для полегшення кулінарного досвіду користувачів, забезпечуючи доступ до високоякісних рецептів з відповідним візуальним поясненням.

Актуальність теми полягає в тому, що вона відповідає сучасним потребам суспільства, спрямованим на оптимізацію часу та покращення якості життя через використання інформаційних технологій. Розробка зручного інструменту для організації рецептів з інтеграцією YouTube може стати важливим кроком у вдосконаленні процесу приготування їжі та зростанні кулінарних навичок широкого кола людей.

Мета роботи. Метою цієї роботи є розробка та впровадження інноваційного інструменту, який дозволить користувачам з легкістю знаходити, організовувати та використовувати рецепти, інтегруючи в них відео-контент з YouTube. Головні цілі проекту включають:

- забезпечення зручного та приємного користувацького досвіду під час використання інструменту;
- підвищення ефективності приготування страв;
- інтеграція з YouTube.

Завдання роботи. Відповідно до обраної теми, робота має такі задачі:

- аналіз конкурентів та існуючих веб-ресурсів;
- вибір мов програмування та технологій;
- розробка зручного дизайну;
- проведення випробувань та тестування продукту.

Об'єкт роботи. Розробка організатора рецептів та кулінарних інструкцій з YouTube, що включає дослідження процесів структурування та оптимізації кулінарного контенту для зручного доступу користувачів.

Предмет роботи. Вивчення та розробка організатора рецептів і кулінарних інструкцій з YouTube, що включає аналіз існуючих рішень, створення зручного інтерфейсу та структурування кулінарного контенту для оптимізації процесу приготування страв.

Методи роботи. Для реалізації організатора рецептів з інтеграцією YouTube була використана мова програмування Python та її бібліотеки, згенеровано API YouTube, мова програмування JavaScript, база даних PostgreSQL, для бекенду використовувала Django, для фронтенду використовувався інтегрований шаблонізатор Django, дизайн було використано з Bootstrap. У ході роботи для ефективного вирішення завдань роботи застосовувалися наукові методи:

- аналізу;
- синтезу;
- систематизації;
- моделювання;
- проекту.

Практичне значення одержаних результатів. В результаті виконання кваліфікаційної роботи був розроблений веб-ресурс для пошуку та перегляду рецептів з YouTube.

Апробація результатів дослідження. Матеріали кваліфікаційної роботи були представлені на Всеукраїнській науково-практичній інтернет-конференції «ІТ екосистема: цифровізація бізнес-процесів в умовах війни», яка відбулася 23-24 листопада 2023 року в Університеті Короля Данила.

Структура роботи. Розділи - 3. Загальний обсяг основної частини - 40 сторінок. Список використаних джерел - 20

РОЗДІЛ 1. АНАЛІЗ ПОТРЕБ ТА КОНКУРЕНТІВ

1.1 Виявлення потреби до організованого засобу з кулінарними рецептами

У світі, де технології невпинно просуваються вперед, з'являються нові можливості для поліпшення нашого повсякденного життя, включаючи таку сферу, як кулінарія. Сучасні технології, зокрема відеоплатформи як YouTube, відкривають безпрецедентні можливості для навчання та обміну знаннями. Проте, попри величезний потенціал, що лежить в основі цих ресурсів, існує проблема організації та ефективного використання контенту. Велика кількість інформації може виявитися важкою для навігації без належних інструментів для її сортування та збереження [1].

Традиційні способи пошуку та зберігання рецептів, такі як кулінарні книги або закладки в інтернет-браузері, не завжди є ефективними в контексті швидкого доступу до потрібної інформації та її практичного використання. Це може призвести до зниження інтересу до кулінарії та обмеження експериментів із новими рецептами через складнощі з пошуком та організацією інформації [2].

Враховуючи це, існує значна потреба в розробці зручного та інтуїтивно зрозумілого організатора рецептів, який б інтегрувався з платформою YouTube. Такий інструмент міг би значно полегшити процес пошуку, збереження та категоризації рецептів, а також надати користувачам доступ до відео-інструкцій безпосередньо через організатор. Це б не лише спростило процес приготування страв, але й зробило його більш привабливим і доступним для широкого кола користувачів, незалежно від їхнього досвіду в кулінарії.

Крім того, організатор рецептів з інтеграцією YouTube міг би забезпечити додаткові можливості для освітнього контенту, включаючи відеоуроки з приготування страв, поради щодо вибору інгредієнтів та методи консервації. Це

б дозволило користувачам не просто слідувати рецептам, а й розвивати свої кулінарні навички, отримуючи знання від шеф-кухарів та інших.

В умовах сучасного ритму життя, коли час на приготування їжі часто є досить обмеженим, наявність такого інструменту може суттєво покращити якість харчування та сприяти розвитку кулінарної культури серед населення.

Отже, розробка організатора рецептів з інтеграцією YouTube є важливим кроком на шляху до вдосконалення доступності та зручності контенту.

1.2 Аналіз існуючих веб-ресурсів з кулінарії

Провівши аналіз вже існуючих веб-ресурсів з тематикою кулінарії, можна сказати, що точного, ідентичного аналогу наразі не існує, проте є веб-сайти, які мають схожий функціонал і можуть бути корисними для користувачів, котрі шукають кулінарні рецепти, та загалом інформацію про приготування їжі.

YouTube (рис.1.1) - платформа на якій авторизовані, або не авторизовані користувачі мають можливість здійснювати пошук за допомогою ключових слів. Вражаюча різноманітність контенту спрямована на цю діяльність, з візуальними інструкціями та доступом до відомих шеф-кухарів. Даний веб-ресурс може бути корисним, як для аматорів, так і для професіоналів [3].

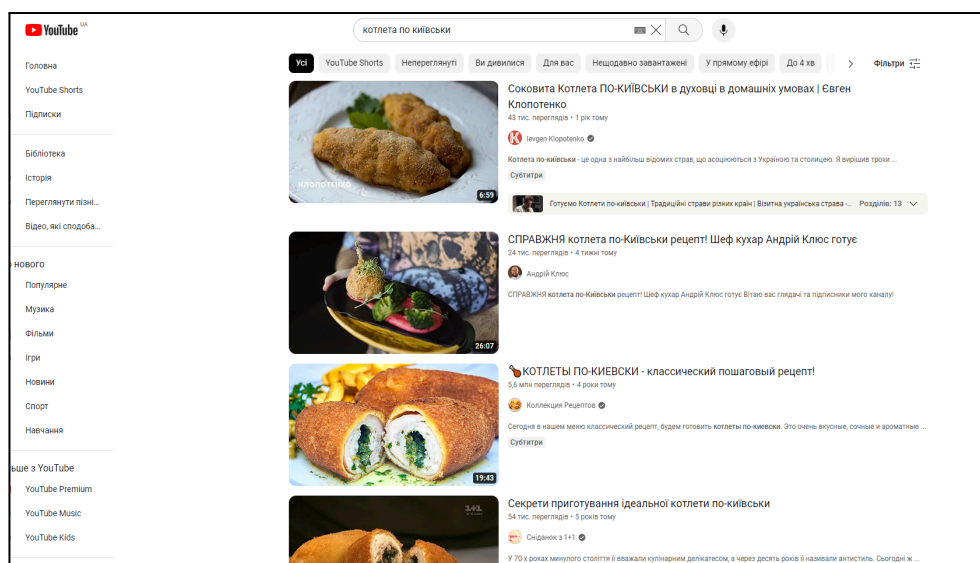


Рисунок 1.1 – Огляд існуючого ресурсу YouTube

Allrecipes (рис.1.2) – є одним з найпопулярніших веб-сайтів з кулінарними рецептами, який з'єднує мільйони любителів готувати по всьому світу. Метою цього сайту є надання користувачам доступу до різноманітних рецептів на будь-який смак і з різних кухонь світу. Авторизовані користувачі мають змогу ділитись фотографіями приготованих рецептів, та оцінювати їх [4].

На основі обраних рецептів Allrecipes дозволяє користувачам створювати шопінг-листи, спрощуючи процес планування покупок та забезпечення необхідними інгредієнтами для приготування страв. Allrecipes пропонує зручний мобільний додаток, який дає можливість користувачам доступати рецепти та інші функції сайту з будь-якого місця, де є інтернет.

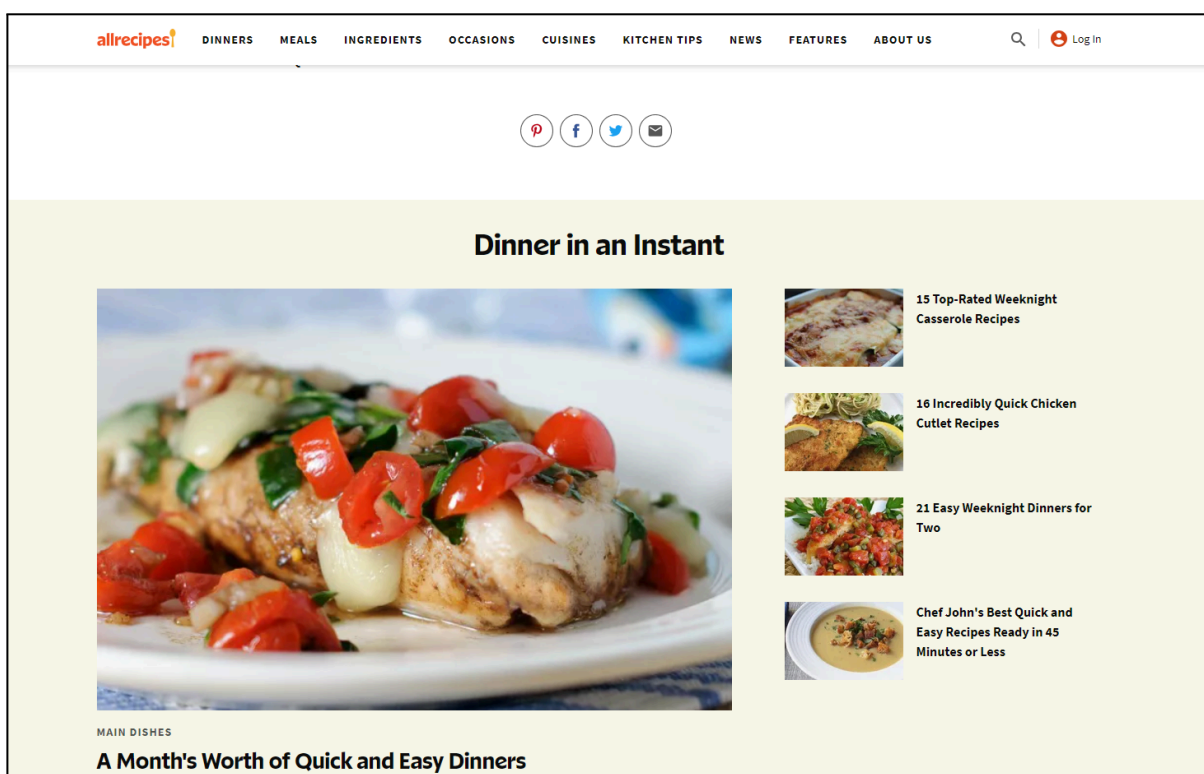


Рисунок 1.2 – Огляд існуючого ресурсу AllRecipes

Cookpad/ua/home (рис.1.3) – це певне онлайн-середовище, де люди можуть ділитися, знаходити і навчатися готувати різноманітні страви, спілкуватися з іншими кулінарними любителями та отримувати натхнення для створення смачної їжі. Cookpad є глобальною платформою для обміну рецептами, яка дозволяє користувачам ділитися своїми кулінарними

відкриттями та знаходити нові ідеї для приготування їжі від інших любителів готувати. Це місце, де можна знайти рецепти на кожен день, а також на особливі випадки. Хоча Cookpad має багато візуальних матеріалів, включаючи фотографії страв, інтеграція з відеорецептами не так розвинена, як на YouTube. Збільшення кількості та якості відеорецептів могло б покращити користувацький досвід [5].

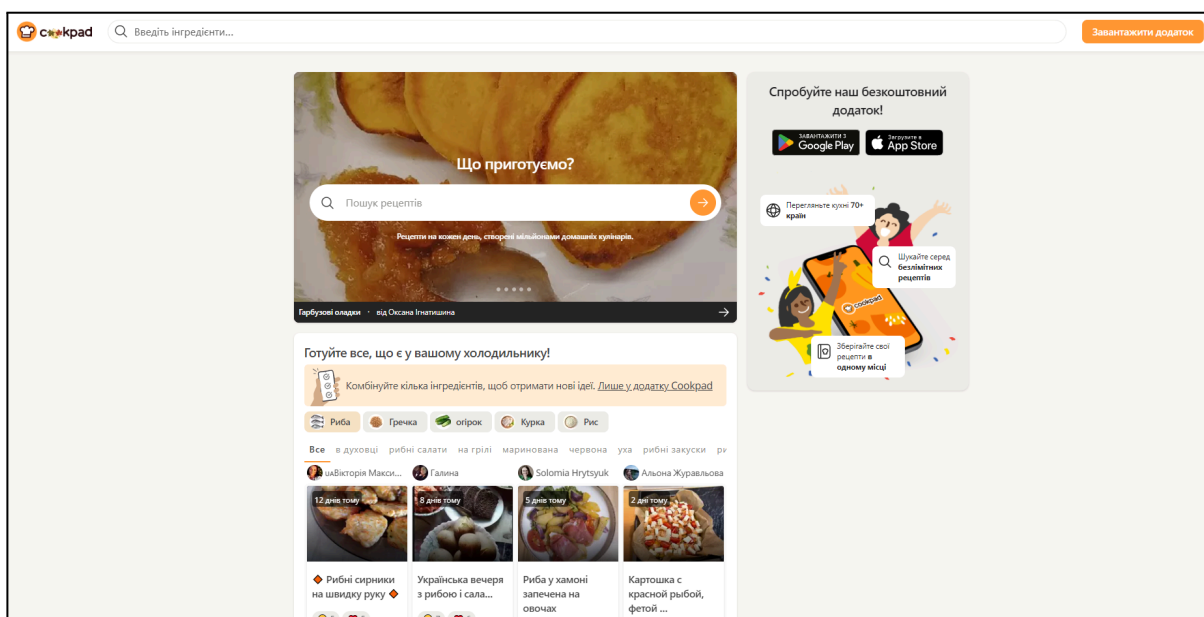


Рисунок 1.3 – Приклад існуючого ресурсу Cookrade.ua/home

Tasty (рис.1.4) – відомий своїми короткими, динамічними відеорецептами, які часто стають вірусними в соціальних мережах, є частиною BuzzFeed. Він здобув широку популярність завдяки легкості виконання рецептів та візуальній привабливості страв. Tasty вдалося створити контент, який одночасно інформативний і розважальний, роблячи процес приготування їжі більш доступним та цікавим. Однією з ключових особливостей Tasty є широкий вибір рецептів, що покривають різноманітні кухні світу та відповідають на різні кулінарні потреби та переваги. Від простих закусок до складних десертів, від традиційних страв до сучасних кулінарних творінь, Tasty пропонує ідеї для кожної події та дієтичних вимог. Це робить платформу улюбленим ресурсом для мільйонів людей по всьому світу. Однак, попри всі переваги, Tasty має певні обмеження. Контент, сфокусований на швидкому та легкому приготуванні, іноді

може залишати недостатньо місця для глибокого занурення в кулінарні процеси та техніки, що може бути мінусом для більш досвідчених кулінарів. Крім того, обмежені можливості для спілкування між користувачами зменшують можливість обміну досвідом та порадами в реальному часі [6].

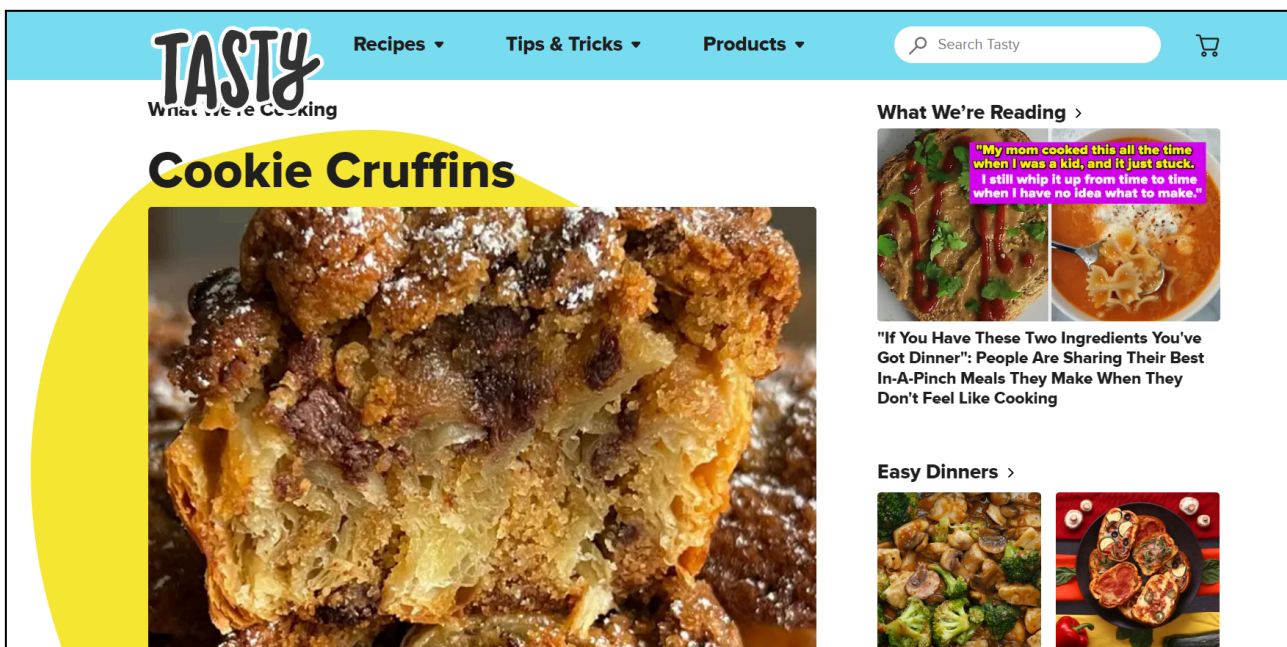


Рисунок 1.4 – Приклад аналогового ресурсу Tasty

1.3 Постановка задачі

У відповідності до вищенаведеного для полегшення пошуку, збереження та використання кулінарних рецептів необхідно створити веб-застосунок, який дозволить користувачам організовувати рецепти з інтеграцією YouTube для перегляду відеоінструкцій. Цей інструмент має на меті об'єднати текстовий та відеоконтент, забезпечуючи зручність та ефективність у використанні кулінарних ресурсів. Відповідно до обраної теми, робота має такі задачі:

- аналіз конкурентів та існуючих веб-ресурсів;
- вибір мов програмування та технологій;
- розробка зручного дизайну;
- інтеграція з YouTube;
- тестування та вдосконалення.

Висновок до розділу 1

У цьому розділі було розглянуто вплив сучасних технологій на кулінарну сферу, зокрема відеоплатформу YouTube, яка відкриває нові можливості для навчання та обміну знаннями. Було визначено проблему організації та ефективного використання кулінарного контенту через велику кількість інформації. Традиційні методи пошуку та зберігання рецептів, такі як кулінарні книги або закладки в браузері, виявилися неефективними в контексті швидкого доступу до потрібної інформації.

Було проведено аналіз існуючих веб-ресурсів з кулінарії, таких як YouTube, Allrecipes, Cookpad і Tasty. Аналіз показав, що хоча ці ресурси мають схожий функціонал, жоден з них не пропонує інтегрованого організатора рецептів з відеоінструкціями. Виявлені обмеження, зокрема в організації контенту та інтерактивності, підкреслюють потребу в розробці зручного та інтуїтивно зрозумілого організатора рецептів з інтеграцією YouTube.

Проведено детальний аналіз конкурентних рішень на ринку, що дозволило виявити сильні та слабкі сторони існуючих веб-ресурсів, розглянуто різні мови програмування та технології, зокрема JavaScript та Python.

Запропоновано ідею створення організатора рецептів, який би полегшував процес пошуку, збереження та категоризації рецептів, а також надавав доступ до відеоуроків безпосередньо через організатор. Такий інструмент міг би значно покращити кулінарний досвід користувачів, зробивши процес приготування їжі більш привабливим і доступним для широкого кола людей, незалежно від їхнього досвіду в кулінарії.

РОЗДІЛ 2. ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ РОЗРОБКИ

2.1 Обґрунтування вибору розробки організатора рецептів з інтеграцією YouTube

У сучасному швидкоплинному світі цифрові технології стрімко розвиваються, відкриваючи нові можливості для вдосконалення нашого життя в різних сферах, включаючи кулінарію. Доступність широкого спектру інформації та навчальних матеріалів в Інтернеті, особливо на відео платформах на кшталт YouTube, відіграє важливу роль у цьому процесі. Це відкриває двері для нових форм навчання та обміну кулінарним досвідом. Однак, перед користувачами постає виклик у вигляді необхідності ефективно організовувати та шукати потрібну інформацію серед безлічі контенту в мережі. Така потреба зумовлює необхідність розробки інноваційних рішень, як-от організатор рецептів з інтеграцією YouTube, який би вирішував цю проблему. Ця інтеграція є стратегічною відповіддю на зростаючий попит на інтерактивні, візуальні та доступні ресурси для вивчення кулінарії [7].

Покращення досвіду навчання є ключовим аспектом такого рішення. Традиційні форми рецептів, як у кулінарних книгах, так і на онлайн-платформах, зазвичай покладаються на текст та статичні зображення. Хоча це ефективно до певної міри, вони не можуть повністю передати нюанси кулінарних технік та динаміку кулінарного процесу. Інтеграція відеоуроків з YouTube безпосередньо в організатор рецептів значно збагачує досвід навчання, пропонуючи користувачам більш повне розуміння через наочну демонстрацію.

Інтеграція з YouTube також вирішує значні бар'єри доступності, з якими багато хто стикається під час навчання кулінарії. Користувачі можуть отримати доступ до величезної кількості кулінарних знань від шеф-кухарів та домашніх кухарів з усього світу, не залишаючи платформу організатора рецептів. Цей

фактор є важливим, враховуючи швидкий темп сучасного життя та зростаючу перевагу навчання в цифрових форматах [8].

Варто зазначити, що організатор рецептів з інтеграцією YouTube може бути реалізований на основі перевірених технологічних рішень. Для розробки такого програмного продукту доцільно використати мову програмування Python, яка вирізняється простотою та читабельністю коду, а також має широкий спектр бібліотек і фреймворків, що значно прискорюють розробку. Для збереження даних може бути обрана надійна та масштабована реляційна база даних PostgreSQL, яка добре інтегрується з фреймворком Django. Вибір Django для розробки серверної частини обґрунтовується його швидкою розробкою веб-додатків, вбудованими механізмами безпеки та зручною взаємодією з базами даних [9].

Для створення зручного та привабливого користувацького інтерфейсу доцільно використати фреймворк Bootstrap, який забезпечує адаптивність, широкий набір компонентів і кросбраузерність. Інтеграція з офіційним YouTube API дозволить реалізувати надійне та функціональне поєднання з відеосервісом, забезпечуючи доступ до великої кількості кулінарного контенту [10].

У підсумку, розробка організатора рецептів з інтеграцією YouTube є прогресивним рішенням, яке відповідає сучасним тенденціям споживання цифрового контенту та задовольняє еволюційні потреби учнів кулінарії. Це обіцяє зробити процес готування більш доступним, захоплюючим та збагачуючим для користувачів, позначаючи значний прогрес у тому, як кулінарна освіта підходить в цифрову епоху.

Інтеграція організатора рецептів з YouTube дозволить користувачам отримати унікальний досвід навчання кулінарії, поєднуючи текстові рецепти з інтерактивними відеоуроками. Це сприятиме кращому засвоєнню знань, підвищенню зацікавленості та залучення аудиторії. Крім того, надання доступу до величезної бази кулінарного контенту на YouTube робить навчання більш інклюзивним та адаптованим до різноманітних стилів сприйняття інформації.

Важливо також зазначити, що така інтеграція є взаємовигідною як для користувачів, так і для творців контенту на YouTube. Направляючи трафік на канали кулінарних блогерів та шеф-кухарів, організатор рецептів підтримує екосистему кулінарного контенту, сприяючи популяризації кулінарного мистецтва та зростанню кількості авторів в цій сфері.

Загалом, розробка організатора рецептів з інтеграцією YouTube є комплексним рішенням, яке відповідає на актуальні потреби сучасних користувачів. Поєднання зручного інструменту для пошуку та збереження рецептів із захоплюючими відеоуроками значно збагатить досвід вивчення кулінарії, роблячи навчання більш доступним, інтерактивним та захопливим. Це дозволить розширити коло зацікавлених у кулінарії, сприяти розвитку навичок готування та популяризувати кулінарне мистецтво серед населення.

У підсумку, розробка організатора рецептів з інтеграцією YouTube є прогресивним рішенням, яке відповідає сучасним тенденціям споживання цифрового контенту та відповідає на еволюційні потреби учнів кулінарії. Це обіцяє зробити готування більш доступним, захоплюючим та збагачуючим для користувачів, позначаючи значний прогрес у тому, як кулінарна освіта підходить в цифрову епоху, та відкриваючи можливості для розвитку кулінарних навичок.

2.2 Обґрунтування вибору технологій та інструментів для розробки веб-ресурсу

Успіх будь-якого програмного проекту значною мірою залежить від правильного вибору технології, мови програмування, алгоритмів, структури та дизайну. Ці фундаментальні аспекти визначають ефективність, масштабованість та привабливість кінцевого продукту. Комплексний підхід до їх опрацювання є запорукою створення успішних, довговічних та затребуваних систем. Від ретельного планування та проектування на початкових етапах залежить весь подальший успіх програмного рішення.

2.3 Обґрунтування вибору мови Python

Одним із ключових рішень під час розробки даного програмного продукту було обрання найбільш підходящої мови програмування. Мова Python була обрана з кількох важливих причин.

По-перше, Python відрізняється простотою та читабельністю свого синтаксису. Це значно полегшує розробку, підтримку та розуміння коду, що є особливо важливим для проектів, які розробляються командою або передбачають подальше супроводження. Лаконічність коду Python дозволяє швидше писати та модифікувати програми, що підвищує ефективність розробки. Крім того, Python має надзвичайно широкий спектр бібліотек та фреймворків, створених спільнотою розробників. Це дає можливість швидко реалізувати різноманітні функціональні можливості, необхідні для вирішення завдань даного проекту. Наприклад, були використані такі перевірені бібліотеки, як Django, Requests, PyTubex та інші. Динамічна типізація та автоматичне керування пам'яттю в Python також сприяють швидкій розробці прототипів та випуску робочих версій програмного забезпечення. Цей фактор суттєво скорочує час розробки та дозволяє оперативно реагувати на зміни вимог [11].

Важливою перевагою Python є його кросплатформеність. Ця мова може використовуватись на різних операційних системах, включаючи Windows, macOS та Linux, що забезпечує універсальність розробленого продукту [12].

Загалом, обрання Python як основної мови програмування для даного проекту обумовлене її простотою, читабельністю коду, широким набором бібліотек, ефективністю розробки та кросплатформеністю. Ці характеристики роблять Python дуже хорошим інструментом для досягнення цілей.

2.4 Обґрунтування вибору бази даних PostgreSQL

Одним з важливих рішень, прийнятих під час розробки, стало обрання бази даних для зберігання даних. Було обрано реляційну базу даних PostgreSQL, що обумовлено кількома ключовими перевагами.

По-перше, PostgreSQL є зрілою, надійною та масштабованою СУБД, здатною ефективно обробляти великі обсяги даних. Це важливо для забезпечення стабільної роботи додатка, особливо у разі значного зростання користувацької бази, що дозволяє підтримувати високу продуктивність.

Крім того, PostgreSQL повністю підтримує стандартний SQL, що спрощує взаємодію з базою даних та інтеграцію з іншими компонентами системи. Наявність широкого спектру вбудованих функцій та можливість додавання нових робить цю СУБД достатньо гнучкою для вирішення завдань проекту [13].

Важливим фактором також стала сумісність PostgreSQL з обраним фреймворком Django для розробки серверної частини. Django має вбудовану підтримку та зручні інструменти для інтеграції з PostgreSQL, що значно спрощує налаштування та взаємодію компонентів програмного продукту.

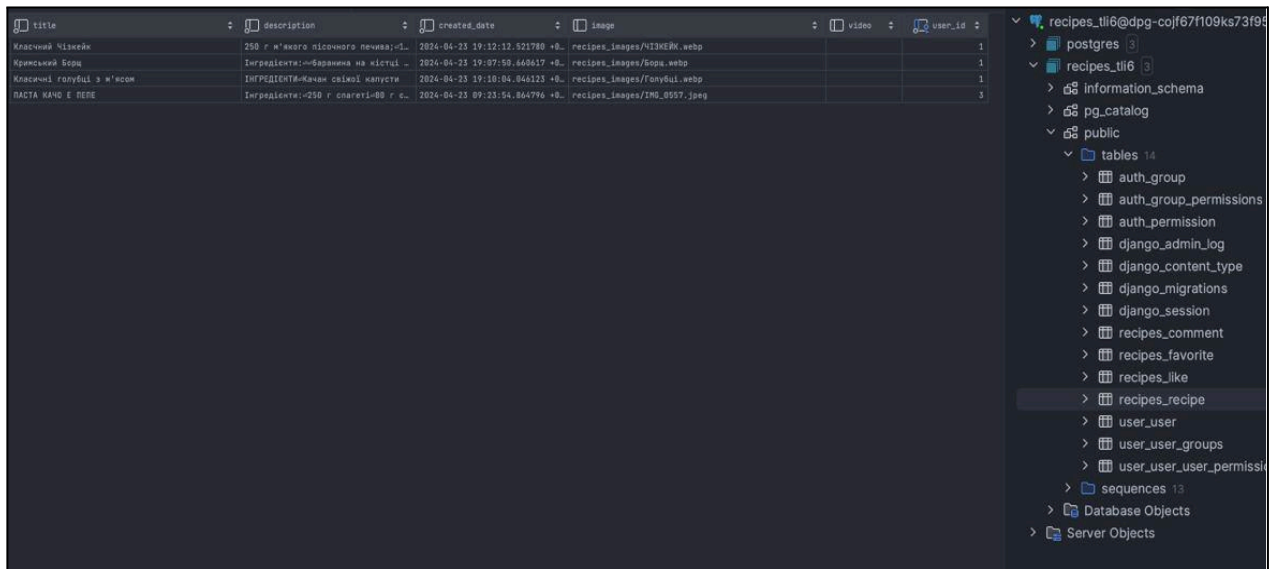
Варто також відзначити, що PostgreSQL є надійною, відмовостійкою та безпечною СУБД, що додатково підвищує стабільність та захищеність розроблюваного додатка, забезпечуючи високу доступність даних.

Отже, вибір PostgreSQL як бази даних для проекту обґрунтовується її надійністю, масштабованістю, підтримкою SQL, розширюваністю та гарною інтеграцією з фреймворком Django. Ці характеристики роблять PostgreSQL ідеальним вибором для забезпечення ефективного зберігання та управління даними в межах даного програмного продукту [14].

2.5 Структура БД

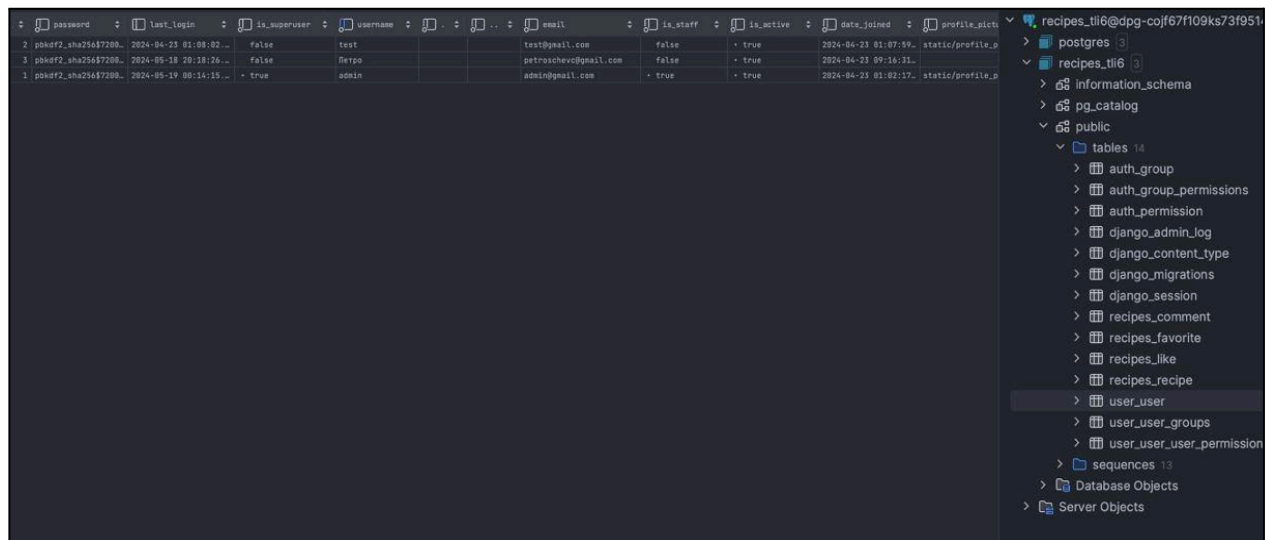
Структура бази даних (рис 2.1, 2.2) проекту була розроблена з урахуванням високих вимог до масштабованості, ефективності та безпеки даних. Основна мета полягала в тому, щоб забезпечити гнучкість системи, яка здатна обробляти великі обсяги даних. Для досягнення цієї мети були ретельно спроектовані відношення між таблицями та визначені відповідні ключі й індекси. Окрім того, були передбачені механізми забезпечення цілісності даних та контролю доступу до інформації в базі даних.

Структура БД включає кілька основних таблиць:



title	description	created_date	image	video	user_id
Класичний Чіккен	250 г м'ясо курочки невеликої породи	2024-04-23 19:12:12.521780 +0...	recipes_images/ТИКЕРК.webp		1
Кремський Борс	Інгредієнти:—Борошна на кістці	2024-04-23 19:07:59.668617 +0...	recipes_images/Борс.webp		1
Класичні голубці з м'ясом	ІНГРЕДІЄНТИ:Качка свіжолі манусти	2024-04-23 19:18:04.046123 +0...	recipes_images/Голубці.webp		1
ПАСТА КАНД Е ПЕРЕ	Інгредієнти:250 г сира сири-80 г с...	2024-04-23 09:23:54.864796 +0...	recipes_images/ІМД_0557.jpg		3

Рисунок 2.1 – Структура бази даних PostgreSQL



password	last_login	is_superuser	username	email	is_staff	is_active	date_joined	profile_picture
pbkdf2_sha256\$7209...	2024-04-23 01:08:02...	false	test	test@gmail.com	false	true	2024-04-23 01:07:59...	static/profile_p...
pbkdf2_sha256\$7209...	2024-05-18 20:18:26...	false	petro	petroshevd@gmail.com	false	true	2024-04-23 09:16:31...	
pbkdf2_sha256\$7209...	2024-05-19 08:14:15...	true	admin	admin@gmail.com	true	true	2024-04-23 03:02:17...	static/profile_p...

Рисунок 2.2 – Структура Бази даних

Таблиця User представляє користувачів системи. Вона містить наступні поля: `username` – ім'я користувача (наслідується від `AbstractUser`), `password` – пароль (наслідується від `AbstractUser`), `email` – електронна пошта (наслідується від `AbstractUser`), `profile_picture` – фото профілю користувача, зберігається у директорії `static/profile_pics/`, `bio` – біографія користувача, `instagram_url` – посилання на Instagram профіль, `telegram_url` – посилання на Telegram профіль.

Таблиця Recipe зберігає інформацію про рецепти. Вона містить наступні

поля: `user` – зовнішній ключ на таблицю `User`, вказує на автора рецепту, `title` – назва рецепту, `description` – опис рецепту, `created_date` – дата створення рецепту, за замовчуванням поточна дата і час, `image` – зображення рецепту, зберігається у директорії `recipes_images/`, `video` – відео рецепту, зберігається у іншій директорії `recipes_videos/`.

Таблиця `Favorite` зберігає інформацію про обрані рецепти користувачів. Вона містить наступні поля: `user` – зовнішній ключ на таблицю `User`, вказує на користувача, який додав рецепт в улюблені, `recipe` – зовнішній ключ на таблицю `Recipe`, вказує на рецепт, `added_on` – дата додавання рецепту в обрані. Унікальність записів забезпечується складеним ключем з полів `user` і `recipe`.

Таблиця `Like` зберігає інформацію про лайки до рецептів. Вона містить наступні поля: `user` – зовнішній ключ на таблицю `User`, вказує на користувача, який поставив лайк, `recipe` – зовнішній ключ на таблицю `Recipe`, вказує на рецепт, `liked_on` – дата, коли лайк був поставлений. Унікальність записів забезпечується унікальним складеним ключем з полів `user` і `recipe`.

Таблиця `Comment` зберігає інформацію про коментарі до рецептів. Вона містить наступні поля: `user` – зовнішній ключ на таблицю `User`, вказує на користувача, який залишив коментар, `recipe` – зовнішній ключ на таблицю `Recipe`, вказує на рецепт, `text` – текст коментаря, `commented_on` – вказує на дату, коли коментар був залишений.

Діаграма зв'язків: один користувач може створити багато рецептів, але кожен рецепт належить тільки одному користувачу (один до багатьох). Один користувач може додати багато рецептів в улюблені, але кожен запис в улюблених належить одному користувачу і одному рецепту (багато до багатьох через таблицю `Favorite`). Один користувач може поставити багато лайків, але кожен лайк належить одному користувачу і одному рецепту (багато до багатьох через таблицю `Like`). Один користувач може залишити багато коментарів, але кожен коментар належить одному користувачу і одному рецепту. Це вказує на зв'язок – один до багатьох.

Ця структура дозволяє зберігати різноманітні дані про рецепти, їх авторів,

а також взаємодію користувачів з цими рецептами (лайки, коментарі, улюблені рецепти), що забезпечує ефективне зберігання та обробку даних.

2.6 Обґрунтування вибору фреймворку Django

Для розробки серверної частини програмного продукту було обрано фреймворк Django. Це рішення обумовлене кількома перевагами Django.

По-перше, Django надає широкий набір інструментів та функціоналу для прискореної розробки веб-додатків. Це включає в себе систему моделей, адміністративну панель, систему маршрутизації та шаблонізації, що суттєво скорочує час розробки додатку чи іншого ресурсу.

Крім того, Django має вбудовані механізми забезпечення безпеки веб-додатків, такі як захист від CSRF, SQL-ін'єкцій та інших поширених вразливостей. Django автоматично генерує токени для захисту від CSRF (Cross-Site Request Forgery), що забезпечує безпеку форм і запитів, запобігаючи несанкціонованим діям від імені користувачів. Це гарантує високий рівень захищеності розроблюваного програмного продукту [15].

Важливою перевагою Django є його масштабованість. Фреймворк добре справляється з обслуговуванням великої кількості користувачів, що є важливим для забезпечення стабільної роботи додатка.

Django також надає зручний та ефективний ORM для взаємодії з базами даних, включаючи підтримку обраної PostgreSQL. Це спрощує роботу з даними.

Ще одним значним фактором стала активна спільнота розробників Django та велика кількість документації. Це полегшує вивчення фреймворку, пошук рішень для типових завдань та подальшу підтримку проекту.

Не менш важливим є те, що Django використовує вбудований шаблонізатор Jinja2, який добре інтегрується з фронтенд-компонентами, такими як Bootstrap. Це спрощує розробку користувницького інтерфейсу.

Отже, вибір Django як фреймворку для розробки серверної частини обумовлений такими факторами, як швидка розробка додатків, вбудовані механізми безпеки, масштабованість, ефективна взаємодія з базами даних,

активна спільнота та гарна інтеграція з фронтендом. Ці характеристики роблять Django надійним інструментом для вирішення завдань проекту [16].

2.7 Обґрунтування вибору фреймворку Bootstrap

Для розробки користувацького інтерфейсу програмного продукту було обрано фреймворк Bootstrap. Це рішення ґрунтується на низці ключових переваг, які Bootstrap надає користувачам.

Одним з найважливіших факторів стала адаптивність та мобільність компонентів Bootstrap. Цей фреймворк забезпечує готові стилі та компоненти, які автоматично підлаштовуються під різні екранні розміри та пристрої. Це критично важливо в сучасних умовах, коли користувачі активно використовують мобільні пристрої для взаємодії з веб-додатками.

Крім того, Bootstrap має надзвичайно широкий набір готових UI-компонентів, таких як навігаційні панелі, форми, модальні вікна тощо. Це значно прискорює розробку користовницького інтерфейсу, дозволяючи зосередитися на реалізації унікальних функціональних можливостей.

Важливою перевагою Bootstrap є його кросбраузерність. Фреймворк коректно працює в основних сучасних веб-браузерах, гарантуючи однорідний вигляд додатка для всіх користувачів [17].

Ще одним ключовим фактором стала легка інтеграція Bootstrap з обраним фреймворком Django. Завдяки вбудованому шаблонізатору Jinja2, Bootstrap інтегрується з Django, що спрощує розробку дизайну та верстку сторінок.

Не менш важливо, що Bootstrap є надзвичайно популярним фреймворком із великою спільнотою розробників. Це забезпечує широку доступність документації, прикладів коду та готових рішень, що полегшує розробку та подальшу підтримку проекту.

Таким чином, вибір Bootstrap як фреймворку для розробки користувацького інтерфейсу ґрунтується на його адаптивності, широкому наборі компонентів, кросбраузерності, легкій інтеграції з Django та високій популярності в спільноті. Ці характеристики роблять Bootstrap оптимальним

рішенням для створення сучасного, користувацького інтерфейсу [18].

2.8 Використання API YouTube

Для реалізації інтеграції з YouTube та отримання даних про відео було обрано використання офіційного YouTube API. Це рішення обумовлене кількома важливими перевагами. По-перше, використання офіційного API, підтримуваного компанією Google, гарантує коректну та стабільну роботу інтеграції. Це забезпечує надійність взаємодії з YouTube сервісом.

Крім того, YouTube API надає широкі можливості для пошуку, отримання метаданих та взаємодії з відео. Цей функціонал повністю задовольняє потреби даного проекту та дозволяє реалізувати всі необхідні інтеграційні сценарії.

Важливою перевагою також є наявність детальної документації YouTube API та великої кількості прикладів реалізації інтеграції. Це значно полегшує розробку, дозволяючи швидко знаходити необхідні рішення.

Не менш важливим аспектом є підтримка різних методів авторизації, включаючи OAuth2. Це забезпечує безпечну та контрольовану взаємодію з сервісом YouTube, гарантуючи надійний захист даних.

Отже, вибір офіційного YouTube API для інтеграції обумовлений його офіційною підтримкою, широким функціоналом, наявністю детальної документації та прикладів, а також підтримкою надійних механізмів авторизації. Ці переваги роблять YouTube API оптимальним вибором для реалізації інтеграції в межах даного програмного продукту [19].

2.9 Use Case Diagram проєкту

Use case diagram (додаток А) є інструментом моделювання, який відображає взаємодію системи з її зовнішнім середовищем. Діаграма показує функціональні вимоги до системи, представлені у вигляді дійових осіб та прецедентів варіантів використання. Вона допомагає зрозуміти, як різні користувачі або інші системи взаємодіють з досліджуваною системою, та які

основні функції повинні бути реалізовані для задоволення потреб цих користувачів або інших систем.

Ця діаграма є важливою частиною процесу розробки програмного забезпечення, оскільки вона забезпечує спільне розуміння функціональності системи між розробниками, аналітиками та замовниками. Вона також використовується для ідентифікації вимог до системи, планування тестування та документації. Використовуючи use case diagram, команди можуть чітко визначити та узгодити обсяг роботи, що підлягає виконанню, та зменшити ризик непорозумінь і помилок у процесі розробки.

Крім того, діаграми випадків використання можуть слугувати основою для створення детальних діаграм, таких як діаграми послідовностей або класів.

У результаті використання use case diagram забезпечує прозорість процесу розробки, що дозволяє ефективніше комунікувати всім учасникам проекту та підвищує якість продукту, гарантуючи, що всі вимоги будуть задоволені.

Ця діаграма варіантів використання (Use Case Diagram) описує основні функції та взаємодії в межах системи організатора рецептів з інтеграцією YouTube, тобто розробленого веб-ресурсу.

Основним актором у діаграмі є "Користувач", який представляє кінцевих користувачів системи. Користувач може виконувати такі ключові прецеденти:

- створення акаунту, тобто користувач може створити особистий обліковий запис в системі;
- вхід на сайт де користувач може авторизуватися в системі за допомогою логіна та пароля, які були створені під час реєстрації;
- пошук рецептів де користувач може виконувати пошук рецептів за різними критеріями, такими як назва, інгредієнт, категорія (фото/відео);
- перегляд рецептів де користувач має можливість переглядати детальну інформацію про рецепти (дата додавання, кількість коментарів);
- публікація рецептів де авторизовані користувачі можуть публікувати власні рецепти в системі, обираючи фото чи відео страви;
- підписки на сайті де користувач може підписуватися на улюблених

авторів рецептів, використовуючи інші соціальні мережі.

Крім того, в діаграмі присутній ще один актор – "Модератор адміністратор", який виконує такі прецеденти:

- редагування контенту коли модератор може редагувати та видаляти публікації користувачів, а також коментарі;
- перевірка відео коли модератор здійснює перевірку, а також підтвердження відеороликів, завантажених користувачами.

Важливою складовою системи є інтеграція з YouTube. Користувачі можуть переглядати та додавати до рецептів відеоматеріали з YouTube, що розширює можливості навчання та візуалізації.

Діаграма також демонструє взаємозв'язки між прецедентами, де одні прецеденти можуть включати або розширювати інші. Наприклад, "Створення акаунту" може включати прецедент "Вхід на сайт", а "Публікація рецептів" може розширювати можливості "Пошуку рецептів".

Загалом, ця діаграма варіантів використання надає цілісне уявлення про функціональні можливості системи організатора рецептів та її ключові взаємодії з користувачами [20].

Висновок до розділу 2

У другому розділі детально розглянуто вибір технологій для розробки веб-ресурсу "Організатор рецептів з інтеграцією YouTube", виходячи з необхідності створення ефективного, безпечного та користувач-орієнтованого рішення. Вибір Python та Django як основи проекту обумовлений їхньою здатністю до швидкої розробки, масштабованості та зручності в підтримці, що забезпечується великою спільнотою і відмінною документацією. PostgreSQL, обрана для управління даними, забезпечує надійність та високу продуктивність завдяки своїй масштабованості та сумісності з Django.

Bootstrap використовується для створення інтуїтивно зрозумілого і адаптивного інтерфейсу, що робить веб-ресурс доступним на будь-яких пристроях і платформах. Інтеграція з YouTube API відкриває доступ до широкої

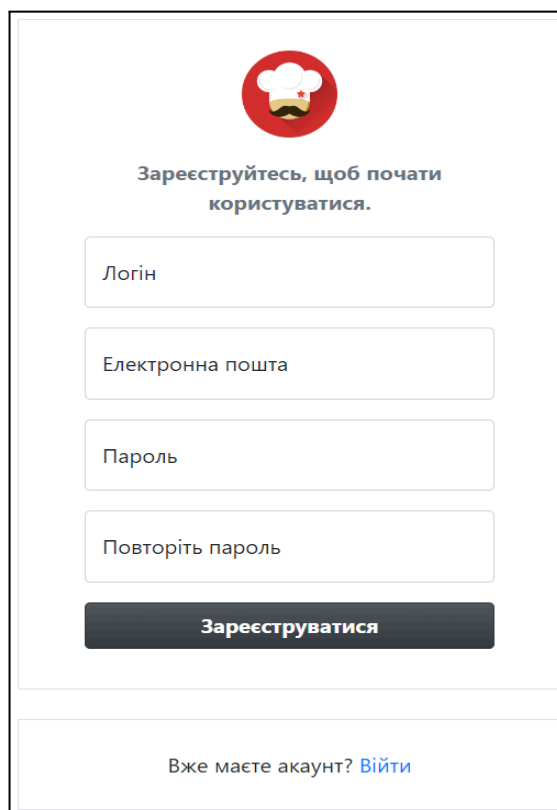
бази відео, розширюючи навчальні можливості користувачів і збагачуючи досвід взаємодії з ресурсом. Ці технології разом створюють міцну основу для веб-ресурсу, який задовольняє сучасні технологічні тенденції та вимоги користувачів, забезпечуючи гнучкість, швидкість та ефективність в кулінарній освіті та обміні досвідом.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБ-РЕСУРСУ

Для старту розробки веб-сайту "Організатор рецептів з інтеграцією YouTube", спочатку необхідно забезпечити наявність усіх необхідних інструментів та ресурсів. Основні кроки включають в себе перевірку доступності всіх інструментів та сервісів, тобто PostgreSQL, Django та його шаблонізатор, а також перевірку наявності необхідних програм на комп'ютері.

3.1 Реалізація форми реєстрації та входу

Для реєстрації користувачів використовується форма, яка складається з декількох полів, що забезпечують зручний інтерфейс для введення даних та відображення помилок(рис.3.1):



Зареєструйтесь, щоб почати користуватися.

Логін

Електронна пошта

Пароль

Повторіть пароль

Зареєструватися

Вже маєте акаунт? [Війти](#)

Рисунок 3.1 – Зображення форми реєстрації користувача

Нижче подано код, який реалізує реєстрацію користувача:

– заголовок форми реєстрації та посилання на головну сторінку:

```
<form action="{% url 'register' %}" method="post" class="bg-white border
py-4 px-5">
  {% csrf_token %}
  <div class="text-center mb-3">
    <a href="{% url 'home' %}">
      
<p class="text-muted fw-bold mt-2">Зареєструйтесь, щоб почати
користування.</p>
</div>
```

Даний код визначає форму для реєстрації користувача з методом POST і вказаним URL для обробки даних. У заголовку форми розміщено логотип з посиланням на домашню сторінку, щоб користувач міг легко повернутися на головну сторінку. Також включено привітальне повідомлення, яке закликає користувача зареєструватися для початку використання сервісу.

Відображення помилок форми :

```
{% if form.non_field_errors %}
  <div class="alert alert-danger">{{ form.non_field_errors }}</div>
{% endif %}
```

Дана частинка коду перевіряє наявність загальних помилок форми, які не прив'язані до конкретних полів. Якщо такі помилки існують, вони відображаються у вигляді червоного повідомлення з класом `alert alert-danger`. Це допомагає користувачу бачити загальні проблеми з введеними даними.

Поле для введення логіну реалізується за допомогою коду нижче:

```
<div class="form-floating mb-3">
  <input type="text" name="username" id="id_username"
    class="form-control{% if form.username.errors %}
is-invalid{% endif %}"
    placeholder="Логін" required value="{{
form.username.value|default_if_none:'' }}">
  <label for="id_username">Логін</label>
```

```

    {% for error in form.username.errors %}
        <div class="invalid-feedback">{{ error }}</div>
    {% endfor %}

```

Поле для введення електронної пошти:

```

<div class="form-floating mb-3">
    <input type="email" name="email" id="id_email"
    class="form-control{% if form.email.errors %} is-invalid{% endif %}"
    </div>
placeholder="Електронна пошта" required value="{{
form.email.value|default_if_none:'' }}">
    <label for="id_email">Електронна пошта</label>
    {% for error in form.email.errors %}
        <div class="invalid-feedback">{{ error }}</div>
    {% endfor %}
</div>
</div>

```

Поле для введення паролю вказано в фрагменті коду поданому нижче:

```

<div class="form-floating mb-3">
    <input type="password" name="password1" id="id_password1"
    class="form-control{% if form.password1.errors %}
is-invalid{% endif %}"
    placeholder="Пароль" required>
    <label for="id_password1">Пароль</label>
    {% for error in form.password1.errors %}
        <div class="invalid-feedback">{{ error }}</div>
    {% endfor %}
</div>
</div>

```

3.2 Головна сторінка

На головній сторінці ресурсу (рис. 3.2) будуть відображені всі необхідні функції для зручного користування.

Перш за все, буде доступний випадючий список, за допомогою якого можна перейти до таких розділів, як "Вподобані", "Обрані" та "Додані рецепти". Це дозволить швидко знаходити потрібні матеріали без зайвих зусиль.

Також на головній сторінці буде розташоване поле пошуку, що допоможе швидко знаходити інформацію на сайті. Окрім того, передбачена вкладка "Відео", яка дозволить здійснювати пошук по відео за допомогою YouTube API.

І, звичайно, користувач матиме можливість перейти до свого профілю, де зможе керувати своїми даними та налаштуваннями.

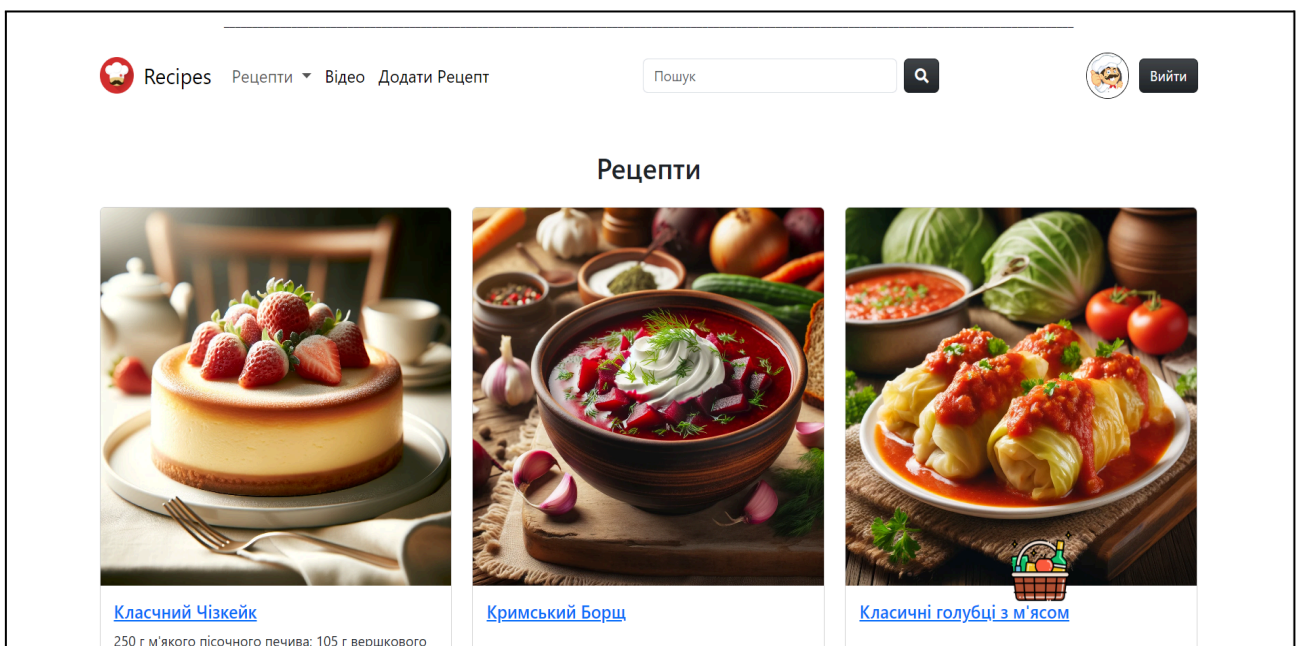


Рисунок 3.2 – Зображення головної сторінки

3.3 Профіль користувача

У профілі користувача (рис.3.3) зберігається основна інформація про його взаємодію з організатором рецептів. Також можна знайти розділ "Вподобані", де відображаються рецепти, які користувач оцінив позитивно та якими поділився з іншими. Розділ "Обрані" містить рецепти, які користувач зберіг для майбутнього використання, вказуючи на те, що він хотів би повторити ці страви. Крім того, у профілі відображається кількість рецептів, доданих користувачем.

Ці додані рецепти будуть розміщені нижче в самому профілі, забезпечуючи легкий доступ до створених користувачем кулінарних інструкцій.

що допомагає створити більш повну картину кулінарних вподобань користувача. Наприклад, буде відображатися дата додавання кожного рецепта, що дозволить користувачу відстежувати, коли саме він створив той чи інший рецепт. Це може бути корисно для планування повторного приготування або для перегляду власного кулінарного розвитку з часом.

Крім того, інші користувачі зможуть залишати коментарі та оцінки під рецептами. Коментарі надають можливість обговорювати рецепти, обмінюватися порадами та рекомендаціями, а також висловлювати подяку або побажання.

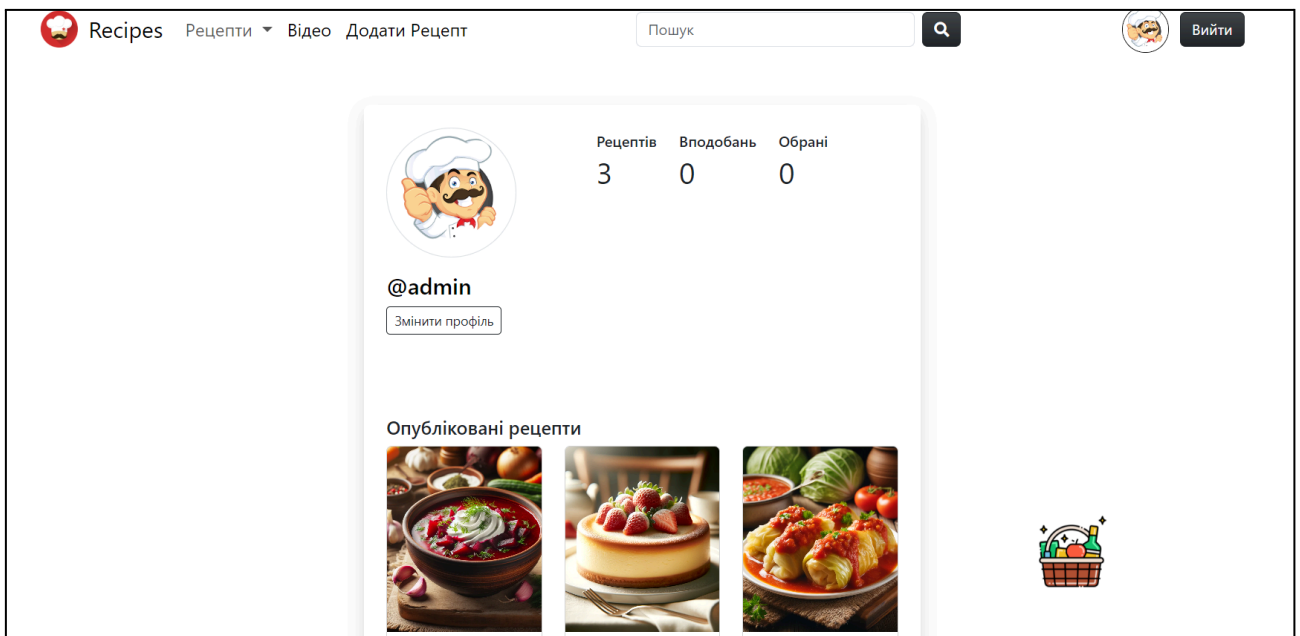


Рисунок 3.3 – Зображення профілю користувача

Частина коду, яка відповідає за секцію з рецептами, які будуть відображатись в профілі:

```
<div class="recipes-section px-4 py-3">
  <h5>Опубліковані рецепти</h5>
  {% if recipes %}
    <div class="row">
      {% for recipe in recipes|slice:"3" %}
        <div class="col-md-4 mb-3">
          <div class="card recipe-card">
```

```
{% if recipe.image %}
```

Зображення рецепту відображається, якщо воно є. Якщо зображення немає, відображається текст "Немає зображення". Є кнопка для переходу до детального перегляду рецепту та кнопка для перегляду всіх рецептів користувача. Якщо рецептів немає, відображається повідомлення:

```

  <div class="recipe-profile-image-placeholder">
    <span>Немає зображення</span>
    <div class="card-body">
<h6 class="card-title">{{ recipe.title }}</h6>
    <a href="{% url 'recipe_detail' recipe.id %}"
class="btn btn-dark bg-gradient">Детальніше</a>
    <a href="{% url 'user_recipes' user.username %}" class="btn
btn-outline-dark">Побачити всі</a>
    <p>Рецептів немає.</p>
```

Також, для того щоб користувачі могли спілкуватися між собою, було прийняте рішення надати можливість залишати посилання на соціальні мережі в їх профілях (рис. 3.4).

Це дозволить користувачам легко знаходити один одного та підтримувати зв'язок, обмінюватися порадами, рецептами та іншою корисною інформацією.

Користувачі можуть легко додати або оновити ці посилання через кнопку “Змінити профіль”, яка дозволяє редагувати інформацію про рецепти, додавати нові зображення, змінювати назви та описи рецептів, а також керувати іншими налаштуваннями профілю. Ця функція забезпечує зручний і простий спосіб тримати свій профіль актуальним і привабливим. Крім того, регулярне оновлення контенту сприяє залученню більшої аудиторії та покращує взаємодію між учасниками спільноти:

Рисунок 3.4 – Зображення функції додавання посилання на соцмережі

Даний функціонал реалізується за допомогою наступного фрагменту коду, поданого на сторінці нижче:

```
<div class="social-links" style="margin-top: 25%;">
  {% if user.instagram_url %}
  <a href="{{ user.instagram_url }}" target="_blank" title="Instagram"
class="social-link-text">
<i class="fab fa-instagram" style="color: #ac2bac;"></i> Instagram
  </a>
  {% endif %}
  {% if user.telegram_url %}
  <a href="https://t.me/{{ user.telegram_url }}" target="_blank"
title="Telegram" class="social-link-text">
    <i class="fab fa-telegram" style="color: #55acee;"></i>
Telegram
  </a>
  {% if user.telegram_url %}
```

3.3 Реалізація підключення YouTube API

В нашому веб-ресурсі підключення YouTube API є дуже важливим аспектом, так як без нього ми не зможемо здійснювати пошук до відео, що і було ключовою метою даного веб-сайту – зібрати все в одне ціле, щоб не переключатися між багатьма ресурсами.

Отже наступний фрагмент коду відповідає за пошук по відеофайлах з ютубу, за переліком ключових слів:

```
import googleapiclient.discovery as yt_api
import random
def scrape_youtube_videos(q):
```

В фрагменті коду поданому нижче здійснюється перевірка на те, чи вхідних запит є порожнім:

```
    if not q:
        key_words = [
            "Борщ", "Вареники", "Голубці", "Деруни", "Сало з часником",
            "Літні салати", "Осінні супи", "Зимові гарячі напої",
            "Весняні десерти",
            "Рецепти на Різдво", "Паска", "Страви на Масницю",
            "Вегетаріанські рецепти", "Сироїдні страви", "Безглютенові
            рецепти",
            "Сніданки за 5 хвилин", "Обіди за 10 хвилин", "Вечері за 15
            хвилин",
            "Як правильно замаринувати м'ясо", "Секрети ідеального
            тіста",
            "Український стріт-фуд", "Гастромаркети України",
            "Домашні наливки", "Узвар", "Квас",
            "Українські пироги", "Домашні торти", "Рецепти печива"
        ]
        key_word = random.choice(key_words)
        q = key_word
```

В рядках поданих нижче здійснюється перевірка на наявність слова “рецепт” в нашому запиті. Якщо слово "рецепт" відсутнє у запиті, то до запиту додається слово "Рецепт" на початку. Це дозволяє забезпечити більш точні результати пошуку:

```
elif not 'рецепт' in q.lower():
    q = f"Рецепт {q}"
```

Підключення до API YouTube, створення запиту, його виконання та отриманні відповідей:

```
youtube = yt_api.build('youtube', 'v3',
developerKey='AIzaSyCJUcq4p69xwn6q9D1d8Ws2qOjIy_pgskQ')
request = youtube.search().list(
    q=q,
    part='snippet',
    type='video',
    maxResults=18,
    regionCode='UA'
response = request.execute()
```

Обробка відповіді та збереження результатів у список:

```
videos = []
for item in response['items']:
    videos.append({
        'title': item['snippet']['title'],
        'description': item['snippet']['description'],
        'url': f"https://www.youtube.com/watch?v={item['id']['videoId']}",
        'thumbnail_url': item['snippet']['thumbnails']['medium']['url']
    })
return videos
```

Використовуючи бібліотеку `googleapiclient.discovery`, функція створює об'єкт `youtube` для взаємодії з YouTube Data API, використовуючи API ключ.

Використовуючи метод `search().list()` з YouTube API, код формує пошуковий запит з параметрами, такими як пошукове слово `q`, тип елементів

'video' (відео), максимальну кількість результатів (18), та регіональний код 'UA' (Україна). Це дозволяє отримати список відео, які відповідають критеріям пошуку. Ключові слова, були додані для того, щоб створити фільтр відеофайлів.

Загалом, цей код дозволяє автоматично збирати дані про відео з YouTube, що стосуються українських кулінарних рецептів, та може бути використаний для створення ресурсу з кулінарними відеогідами або як частина більш широкої платформи кулінарного контенту. Автоматизація процесу збору даних значно знижує необхідність ручного пошуку та збирання інформації, що дозволяє заощадити час та ресурси. Крім того, цей підхід гарантує, що інформація буде постійно оновлюватися, відображаючи останні та найпопулярніші кулінарні відео на YouTube, забезпечуючи доступ до найактуальніших рецептів.

При натисканні на кнопку “Дивитись відео” нас переадресовує в веб-ресурс YouTube на сторінку з даним відео.

Приклад роботи даного поля, можна побачити нижче на рисунку 3.5:

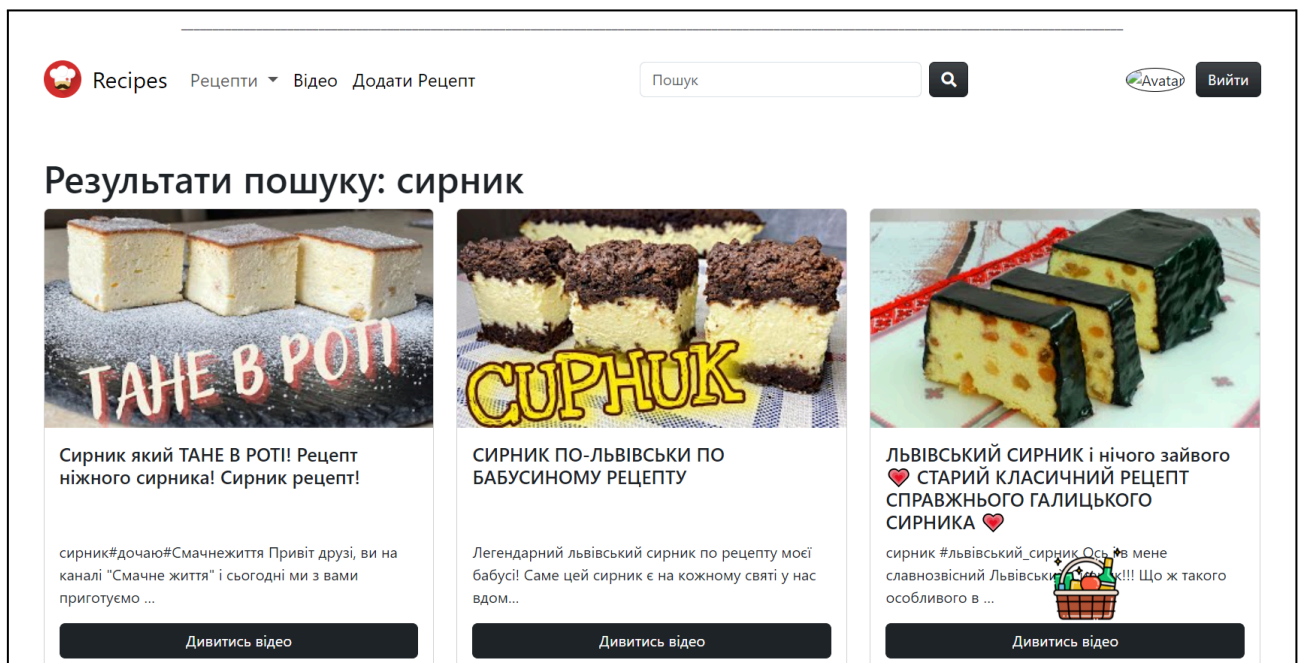


Рисунок 3.5 – Приклад пошуку через YouTube API

3.4 Реалізація функціоналу “Нотаток”

Одним із цікавих елементів інтерфейсу — це нотатки або продуктова

корзина, де користувачі можуть записувати список продуктів, необхідних для приготування обраного рецепту.

Ця функція веб-сайту є особливо корисною, оскільки дозволяє користувачам зберегти та зручно переглянути список потрібних інгредієнтів під час покупок або готування. Особливістю цих нотаток є те, що при оновленні або повторному відкритті сторінки список інгредієнтів залишається збереженим. Це значно покращує користувацький досвід, оскільки не потрібно щоразу заново вводити або запам'ятовувати список продуктів.

Код поля для нотаток:

```
<div id="noteContainer">
    <button id="showNoteInput" class="floating-button"></button>
    <div id="noteInput">
        <textarea id="noteContent" placeholder="Введіть вашу нотатку
тут..." rows="4" cols="50"></textarea>
```

Функції управління видимістю та збереження даних:

```
document.getElementById("showNoteInput").onclick = function () {
    var x = document.getElementById("noteInput");
    if (x.style.opacity === "1") {
        x.style.opacity = "0";
        x.style.visibility = "hidden"
        x.style.opacity = "1";
        x.style.visibility = "visible";
    }
    document.getElementById("noteContent").oninput = function () {
        localStorage.setItem("noteContent", this.value);
    }
    window.onload = function () {
        var savedNote = localStorage.getItem("noteContent");
        if (savedNote) {
            document.getElementById("noteContent").value = savedNote;
        }
    }
}
```

Перевірка видимості поля вводу при завантаженні сторінки і коригування його стану:

```

var noteInputVisibility = localStorage.getItem("noteInputVisibility");
if (noteInputVisibility === "visible") {
    document.getElementById("noteInput").style.opacity = "1";
document.getElementById("noteInput").style.visibility = "visible";

```

Збереження стану видимості поля вводу:

```

document.getElementById("showNoteInput").addEventListener("click",
function () {
    var visibility = document.getElementById("noteInput").style.visibility;
    localStorage.setItem("noteInputVisibility", visibility);

```

В кодї поданому вище присутні декілька основних функцій,наприклад, перемикання видимості поля вводу,який відповідає за закривання і відкривання цих ж нотаток за допомогою кнопки. В нашому випадку під кнопку підлаштована відмальована картинка корзини з продуктами.

Також присутня функція збереження стану видимості поля вводу,яка дозволяє зберегти стан видимості нотаток після перезавантаження сторінки,для того щоб користувач мав змогу продовжити роботу з тим самим інтерфейсом,що був перед перезавантаженням сторінки. Збереження стану видимості поля вводу реалізовано за допомогою механізмів веб-браузера, таких як localStorage або sessionStorage. Це дозволяє зберігати інформацію про стан інтерфейсу на стороні клієнта, тобто без необхідності надсилати дані на сервер. Сам вигляд “Нотаток” можемо побачити на рисунку 3.6 поданому нижче:

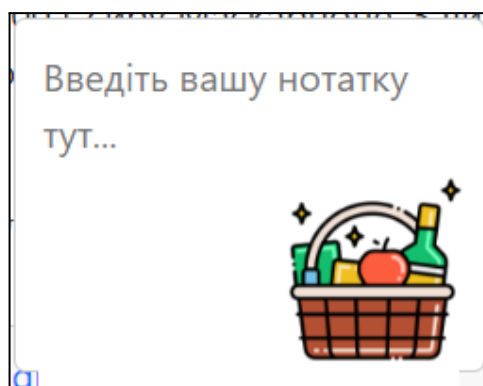


Рисунок 3.6 – “Нотатки”

3.5 Реалізація змоги редагування рецептів

Для того, щоб забезпечити функціональність редагування рецептів у проєкті, потрібно мати кілька компонентів: модель рецепту, форму для редагування, представлення (view), яке обробляє запити на редагування, та URL-конфігурацію для маршрутизації запитів.

Ось створена форма для редагування рецепту прописана в коді:

```
<section class="py-5">
  <div class="container">
    <div class="row d-flex align-items-center
justify-content-center">
      <div style="max-width:420px;">
        <form method="post" enctype="multipart/form-data"
class="bg-white shadow-sm p-4">
          {% csrf_token %}
          <div class="text-center mb-4">
            {% if recipe.image %}
              
            <h2 class="text-muted fw-bold">Редагування рецепту</h2>
```

Частина коду подана вище створює форму редагування рецепту, яка розміщена всередині секції з відступами та контейнера для центрування. Рядок із класами `d-flex align-items-center justify-content-center` забезпечує вертикальне та горизонтальне вирівнювання вмісту;

```
</div>
  {% for field in form %}
    <div class="mb-3">
      {% if field.widget.input_type != 'hidden' %}
<label for="{{ field.id_for_label }}" class="form-label">{{ field.label
}}</label>
      {% endif %}
```

Блок коду поданий вище створює контейнер для кожного поля форми,

додаючи відступ знизу, а також якщо тип поля не є прихованим до додається мітка 'label';

```

    {% if field.widget.input_type == 'file' %}
        <div class="input-group">
            <input type="text" class="form-control" placeholder="{{ field.label }}"
            readonly>
            <button class="btn btn-outline-secondary" type="button"
            onclick="document.getElementById('{{ field.id_for_label
            }}').click();">Вибрати файл</button>
                {{ field }}
                {% if field.help_text %}
                    <small class="form-text text-muted">{{
            field.help_text }}</small>
                {% endif %}

```

Наведений рядками вище код описує обробку відображення полів форми в залежності від їх типу. Якщо поле є полем для завантаження файлу, створюється група вводу з текстовим полем, яке показує ім'я вибраного файлу, і кнопкою, що відкриває вікно вибору файлу;

```

    {% if field.errors %}
        <div class="alert alert-danger mt-2">{{ field.errors }}</div>
    {% endif %}
    {% endfor %}
    <div class="text-center">
        <button type="submit" class="btn btn-dark fw-bold w-100
            mt-3">Зберегти зміни</button>
        <a href="{% url 'recipe_detail' recipe.id %}" class="btn
            btn-outline-secondary mt-2 w-100">Скасувати</a>
    </div>
</form>

```

Наведена вище частина коду обробляє відображення помилок для кожного поля форми, показуючи їх у вигляді червоного повідомлення, якщо вони існують. Після завершення циклу по полях форми, додається контейнер з

кнопками: кнопка для збереження змін і кнопка для скасування, яка веде до детального перегляду рецепту.

Також для кращого візуального сприйняття можна поглянути на рисунок 3.7, який поданий нижче:

Редагування рецепту

Назва
Кримський Борщ

Опис
Інгредієнти:
баранина на кістці – 500 г,
цибуля – 2 шт,
перець червоний свіжий – 1 шт,
помідор свіжий – 1 шт або в'ялених – 5-6 шт,
кукурудза – качан,
айва – 1 шт,
червоний буряк – 1 шт,

Фото Наразі: recipes_images/67e758b-vadimvasenin-depositphotos.jpg Очистити

Змінити:
Вибрати файл Файл не вибрано

Рисунок 3.7 – Форма редагування рецепту

Форма автоматично заповнюється існуючими даними рецепту завдяки контексту, переданому з Django views. Це забезпечує, ввидимість користувачам поточних значень полів, які вони збираються редагувати:

```
<form method="post" enctype="multipart/form-data" class="bg-white shadow-sm p-4">
    
    {% for field in form %}
        {{ field }}
    {% endfor %}
```

Також що є важливим у конфіденційності, редагувати рецепти можуть тільки авторизовані користувачі, для цього використовувалась функція декоратор `@login_required`, якщо ж користувач не авторизований, його перенаправляє на сторінку входу.

3.6 Створення форми додавання рецептів

Форма включає кілька полів, які користувачі можуть заповнити, щоб поділитися своїм рецептом на сайті. Вона має простий і чистий дизайн, що сприяє легкому введенню даних. Вона містить такі компоненти:

- **назва:** поле для вводу назви рецепту. Це текстове поле дозволяє користувачам ввести назву свого кулінарного творіння;
- **опис:** велике текстове поле для детального опису рецепту. Тут можна описати процес приготування, необхідні інгредієнти та інші деталі;
- **фото:** кнопка для завантаження фотографії страви. "Вибрати файл" дозволяє користувачам завантажити зображення зі свого пристрою;
- **відео:** аналогічно полю для фото, це поле дозволяє завантажити відео, що демонструє процес приготування або презентує готову страву;
- **кнопка "Додати":** кнопка в кінці форми, яка дозволяє користувачу відправити всю інформацію на сервер.

Ця кнопка активує процес збереження даних в базу даних, рисунок 3.8.

Реалізація даної форми в коді:

```
<section class="py-5">
  <div class="container">
    <div class="row d-flex align-items-center
justify-content-center">
      <div style="max-width:420px;">
```

Створення форми додавання рецептів:

```
<form method="post" enctype="multipart/form-data"
class="bg-white border py-4 px-5">
```

```

{% csrf_token %}
<div class="text-center mb-3">
  <a href="{% url 'home' %}">
    
  <p class="text-muted fw-bold mt-2">Додайте ваш
рецепт</p>
{% for field in form %}

```

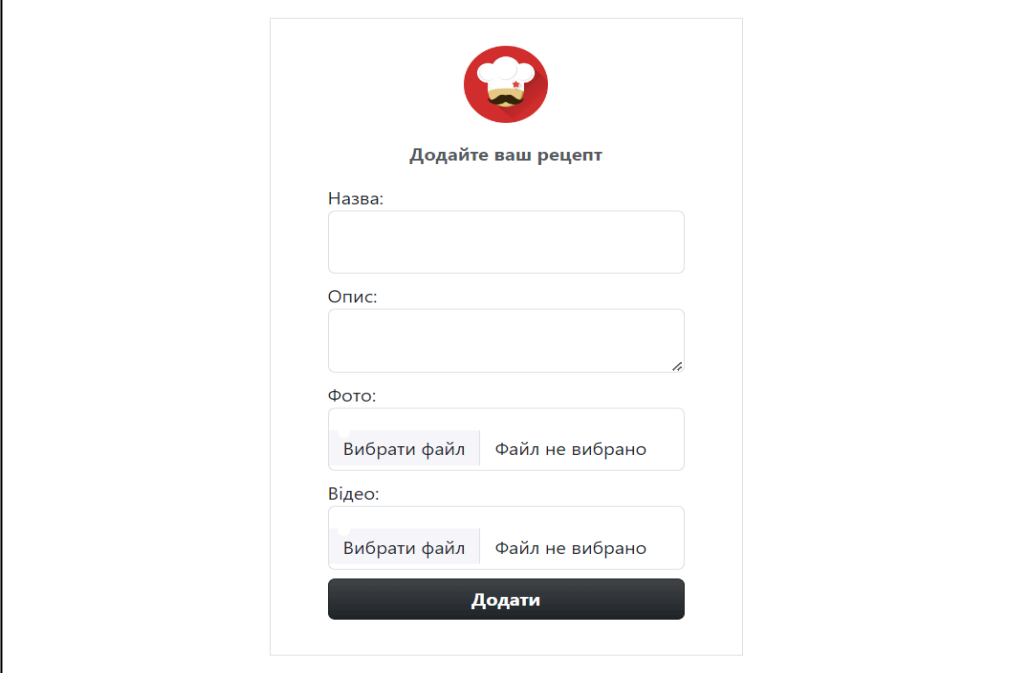
Виведення полів з форми можна поглянути в коді нижче, де всі поля форми, виводять їх із відповідними мітками та допоміжним текстом. Якщо є помилки, вони відображаються у вигляді повідомлення про помилку. Це забезпечує зручний інтерфейс та допомагає виправляти помилки вводу:

```

<div class="mb-2">
  {{ field.label_tag }}
  {% if field.is_hidden %}
    {{ field }}
  {% else %}
    <div class="form-floating">
      {{ field }}
    <label for="{{ field.id_for_label }}">{{ field.help_text }}</label>
    {% endif %}
    {% if field.errors %}
      <div class="alert alert-danger">{{ field.errors }}</div>
    {% endfor %}
    <div class="mb-2">
      <button class="btn btn-dark bg-gradient fw-bold
w-100 bg-gradient" type="submit">Додати</button>
    {% endfor %}
</div>
<div>
  <div>
    <div>

```

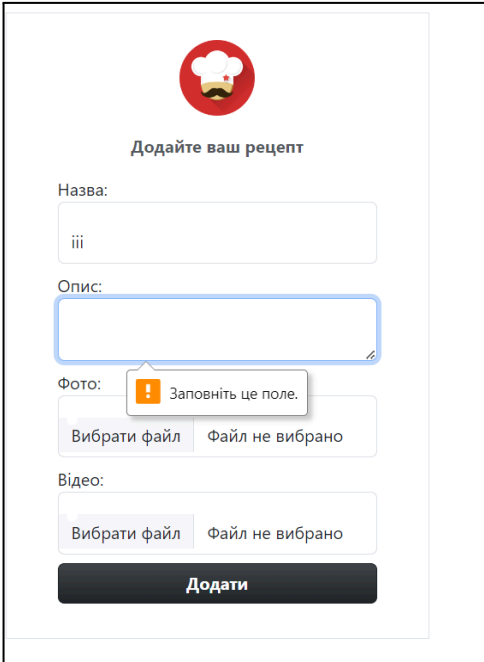

Зображення яке демонструє роботу коду подано нижче (рис.3.8):



The screenshot shows a web form for adding a recipe. At the top is a red circular icon with a white chef's hat. Below it is the title "Додайте ваш рецепт". The form contains several input fields: "Назва:" (Name) with an empty text box; "Опис:" (Description) with an empty text area; "Фото:" (Photo) with a "Вибрати файл" (Select file) button and the text "Файл не вибрано" (File not selected); and "Відео:" (Video) with a "Вибрати файл" (Select file) button and the text "Файл не вибрано" (File not selected). At the bottom is a dark grey button labeled "Додати" (Add).

Рисунок 3.8 – Форма додавання рецепту

Важливим аспектом є те, що додати щось без опису неможливо, тому що при натисканні "Додати" сайт запрошує заповнити поле, демонстрацію цієї дії можна побачити на рисунку 3.9 поданому нижче:



This screenshot shows the same form as Figure 3.8, but with a validation error. The "Назва:" field contains the text "iii". The "Опис:" field is highlighted with a blue border. A tooltip with an orange exclamation mark icon and the text "Заповніть це поле." (Fill in this field.) is displayed over the "Фото:" field. The "Відео:" field is also highlighted with a blue border. The "Додати" button is visible at the bottom.

Рисунок 3.9 – Приклад роботи поля "Додавання рецепту"

Висновки до розділу 3

Безсумнівно, третій розділ є ключовим для розуміння процесу розробки веб-ресурсу "Організатор рецептів з інтеграцією YouTube". В ньому детально описані різноманітні аспекти реалізації цього проекту, від базових елементів, таких як форми реєстрації та входу, до більш складних функцій, як інтеграція з YouTube API та система нотаток для зручності користувачів.

Одним з найважливіших компонентів є профіль користувача, який став центральним місцем для відображення персональної інформації, вподобаних та обраних рецептів, а також дозволив користувачам ділитися посиланнями на свої сторінки в соціальних мережах. Це не лише зробило веб-сайт більш зручним у використанні, а й заохотило користувачів до взаємодії та обміну контентом.

Інтеграція з YouTube API була ключовим кроком у наданні користувачам доступу до величезної бази відеоматеріалів, пов'язаних з кулінарними рецептами. Ця функція дозволяє об'єднати текстовий та відеоконтент в одному місці, створюючи комплексний ресурс для любителів кулінарії.

Крім того, реалізація зручної системи нотаток для збереження списків інгредієнтів підвищила практичність веб-сайту та полегшила процес планування покупок та приготування страв.

Нарешті, можливість редагувати та додавати нові рецепти забезпечила інтерактивність та залучення користувачів до створення контенту. Це не лише збагатило базу рецептів на веб-сайті, а й заохотило спільноту ділитися своїми кулінарними ідеями та досвідом.

Загалом, цей розділ демонструє ретельну роботу та продуману реалізацію різноманітних функцій, що зробили веб-ресурс "Організатор рецептів з інтеграцією YouTube" зручним, корисним та привабливим для користувачів.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було створено веб-застосунок "Організатор рецептів з інтеграцією YouTube", який сприяє популяризації кулінарії та надає зручну можливість користувачам онлайн відкривати нові рецепти, дивитися відеоуроки від майстрів кулінарного мистецтва та ділитися власними рецептами. Цей інноваційний ресурс не лише полегшує взаємодію між кулінарами-аматорами та професіоналами, а й сприяє культурному обміну через кулінарні традиції.

Проект включав наступні етапи розробки:

- аналіз існуючих рішень і визначення потреб користувачів: проведено аналіз сучасних кулінарних веб-ресурсів, таких як YouTube, Allrecipes, Cookpad і Tasty, для визначення їхніх сильних та слабких сторін;

- вибір технологій для реалізації проекту: Python та Django стали основою завдяки їхній швидкості розробки, масштабованості та зручності в підтримці. PostgreSQL обрано для управління даними, забезпечуючи надійність та високу продуктивність. Для створення адаптивного інтерфейсу було використано Bootstrap, а також для ширшого вибору відео було інтегровано YouTube API;

- дизайн та реалізація користувацького інтерфейсу, який є інтуїтивно зрозумілим і простим у використанні, надаючи доступ до основних функцій, таких як пошук рецептів, перегляд відео, реєстрація та управління профілем;

- розробка ширшої функціональності користувацьких нотаток для зберігання інгредієнтів та планування покупок.

Загалом, розробка веб-ресурсу, демонструє ретельну роботу та продуману реалізацію різноманітних функцій, що зробили його зручним, корисним та привабливим для широкого кола користувачів. Використання сучасних технологій дозволило створити ефективний, безпечний та користувач-орієнтований ресурс, який відповідає сучасним тенденціям та вимогам у сфері кулінарії.

Таким чином, веб-ресурс "Організатор рецептів з інтеграцією відео з YouTube" став вдалим прикладом поєднання сучасних технологій, ретельного планування та клієнтоорієнтованого підходу, що зробило його корисним та популярним серед шанувальників кулінарії.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

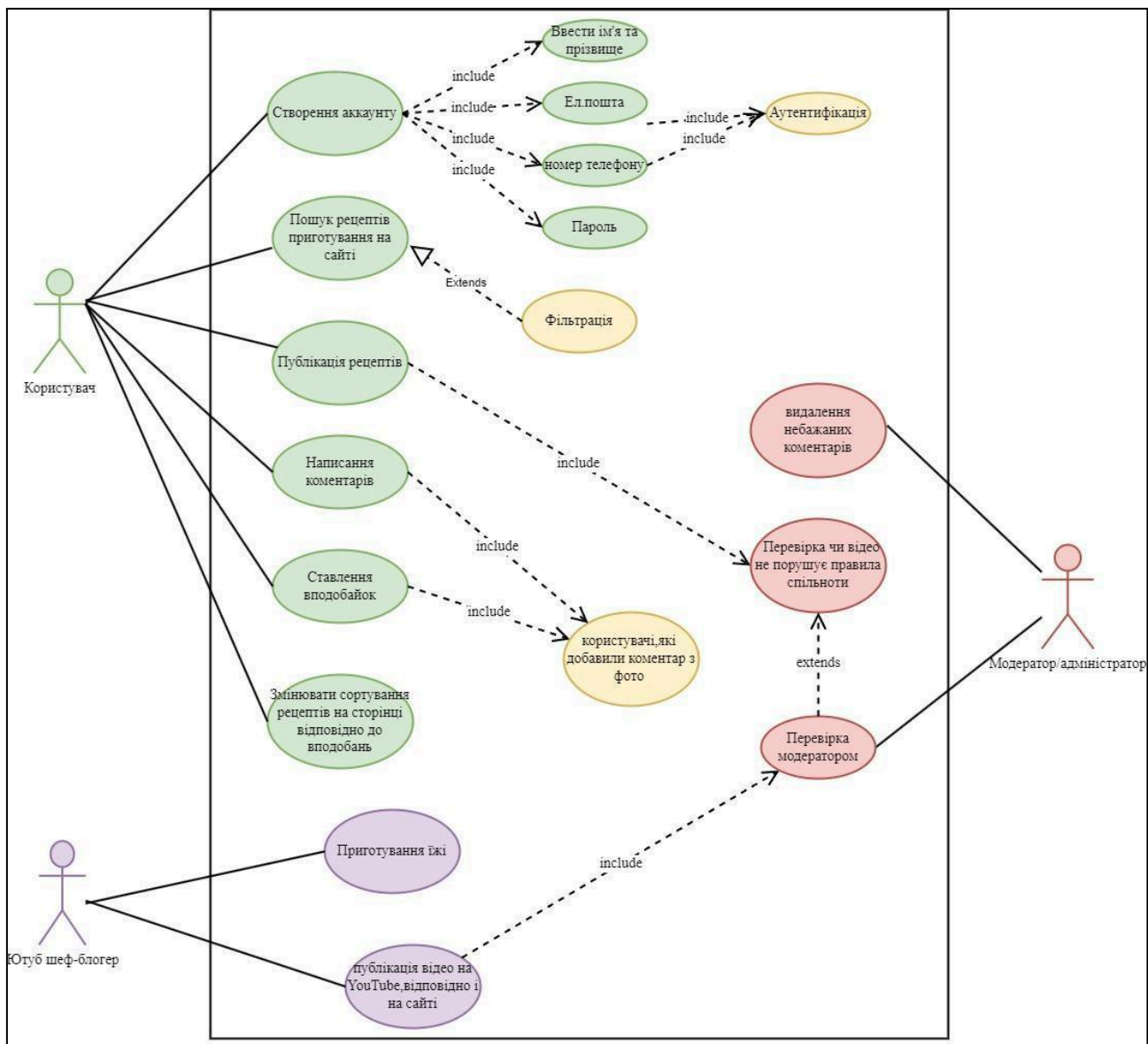
1. Chefvpu75. Кулінарна історія веб-сайт. URL: https://chefvpu75.blogspot.com/p/blog-page_30.html (Дата звернення: 21.03.2024)
2. Pixelinform. (n.d.). Методи які спростять приготування та зберігання рецептів веб-сайт. URL: <https://pixelinform.com/uncategorized/5-kulinarnyh-lajfhakiv-yaki-sprostyat-prygotuvannya-ta-zberigannya-produktiv/> (Дата звернення: 23.03.2024)
3. YouTube. (n.d.). веб-сайт. URL: <https://www.youtube.com/> (Дата звернення: 25.03.2024)
4. Allrecipes. веб-сайт. URL: <https://www.allrecipes.com/> (Дата звернення: 27.03.2024)
5. Cookpad. Головна сторінка. веб-сайт. URL: <https://cookpad.com/ua/homepage> (Дата звернення: 01.04.2024)
6. Tasty. веб-сайт. URL: <https://tasty.co/> (Дата звернення: 03.03.2024)
7. Vinmost. У світі кулінарії є щось більше, ніж просто їжа, це справжнє мистецтво створення. веб-сайт. URL: <https://vinmost.com.ua/articles/u-sviti-kulinariyi-ye-shchos-bilshe-nizh-prosto-yizha-t-se-spravzhnye-mystetstvo-stvorennya> (Дата звернення: 05.04.2024)
8. Elit-Web. Скільки коштує реклама на YouTube. веб-сайт. URL: <https://elit-web.ua/ua/blog/skolko-stoit-reklama-na-youtube> (Дата звернення: 07.04.2024)
9. Foxminded. PostgreSQL: що це?. веб-сайт. URL: <https://foxminded.ua/postgresql-shcho-tse/> (Дата звернення: 09.04.2024)
10. Google Developers. YouTube API v3. веб-сайт. URL: <https://developers.google.com/youtube/v3> (Дата звернення: 11.04.2024)
11. Python. Tutorial. веб-сайт. URL: <https://docs.python.org/uk/3/tutorial/index.html> (Дата звернення: 13.04.2024)

12. Ruslan.rv.ua. Модулі в Python. веб-сайт. URL:
<http://ruslan.rv.ua/python-starter/modules/modules.html> (Дата звернення:
15.04.2024)
13. Data Life UA. Postgres is eating the database world. веб-сайт. URL:
<https://data-life-ua.com/db/postgres-is-eating-the-database-world/> (Дата звернення:
17.04.2024)
14. PostgreSQL. Documentation. веб-сайт. URL:
<https://www.postgresql.org/docs/> (Дата звернення: 19.04.2024)
15. Django. Документація. веб-сайт. URL:
<https://docs.djangoproject.com/en/5.0/> (Дата звернення: 21.04.2024)
16. Webcase. Розробка веб-додатків з використанням Python і Django.
веб-сайт. URL:
<https://webcase.com.ua/uk/blog/razrobotka-veb-prilozhenij-s-ispolzovaniem-python-i-django/> (Дата звернення: 23.04.2024)
17. W3Schools. Bootstrap. веб-сайт. URL:
<https://www.w3schools.com/bootstrap/> (Дата звернення: 25.03.2024)
18. Bootstrap.GettingStarted.веб-сайтURL:
<https://getbootstrap.com/docs/5.3/getting-started/introduction/> (Дата звернення:
27.04.2024)
19. Google Developers. YouTube IFrame Player API. веб-сайт. URL:
https://developers.google.com/youtube/iframe_api_reference (Дата звернення:
29.04.2024)
20. MindOnMap. What is a UML Use Case Diagram?. веб-сайт. URL:
<https://www.mindonmap.com/uk/blog/what-is-a-uml-use-case-diagram/> (Дата
звернення: 09.05.2024)

ДОДАТКИ

Додаток А

Use Case Diagram для мого веб-ресурсу





метадані

Заголовок

Розробка організатора рецептів з інтеграцією YouTube

Автор

Батіг В. М. Науковий керівник / Експерт

підрозділ

King Danylo University

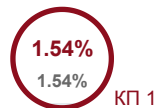
Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про **МОЖЛИВІ** маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		0
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		10

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

**25**

Довжина фрази для коефіцієнта подібності 2

7057

Кількість слів

56782

Кількість символів

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Колір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	Колір тексту
1	http://repository.ukd.edu.ua/bitstream/handle/123456789/396/%D0%94%D0%B8%D0%BF%D0%BB%D0%BE%D0%BC%D0%BD%D0%B0%20%D0%A1%D1%82%D1%80%D1%96%D0%BB%D0%B5%D1%86%D1%8C%D0%BA%D0%B8%D0%B9.pdf?sequence=1	16	0.23 %
2	http://repository.ukd.edu.ua/bitstream/handle/123456789/392/%D0%9A%D0%A0%20%D0%9F%D0%B0%D1%88%D0%BD%D0%B8%D0%BA%20%D0%90.%20%D0%9F_.pdf?sequence=1	12	0.17 %
3	Дослідження моделей інтерактивного електронного портфоліо графічного дизайнера та ілюстратора.pdf 12/20/2021 Odessa National Polytechnic University (ІПТДМ, Каф. ІТІМ)	12	0.17 %