

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: **«Веб-додаток для автоматизації документообігу з
використанням чат-боту»**

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
(код, найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)

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

Микола МИРОНЧУК
(ім'я, ПРІЗВИЩЕ здобувача)

(підпис)

Виконав: здобувач вищої освіти група ІСД-42

Микола МИРОНЧУК

(ім'я, ПРІЗВИЩЕ)

Керівник

PhD

Віктор САГАЙДАК

(ім'я, ПРІЗВИЩЕ)

Рецензент:

(ім'я, ПРІЗВИЩЕ)

Київ 2026

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти бакалавр

Спеціальність 126 Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедрою ІСТ

Каміла СТОРЧАК

“ _____ ” _____ 2026 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Мирончуку Миколі Володимировичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Веб-додаток для автоматизації документообігу з використанням чат-боту

керівник кваліфікаційної роботи: Віктор САГАЙДАК PhD

(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від “20” лютого 2026 р. № 51

2. Строк подання кваліфікаційної роботи «30» травня 2026 р.

3. Вихідні дані кваліфікаційної роботи:

1. Принципи побудови та функціонування систем електронного документообігу.
2. Особливості розробки веб-додатків на основі сучасних веб-технологій.
3. Принципи реалізації серверної частини веб-додатку з використанням фреймворку Flask.
4. Методи інтеграції чат-ботів у веб-додатки для автоматизації взаємодії з користувачами.
5. Науково-технічна література та документація сучасних інформаційних систем і веб-технологій.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1. Аналіз предметної області документообігу та сучасних систем його автоматизації.
2. Дослідження можливостей використання веб-технологій та чат-ботів у інформаційних системах.
3. Проєктування архітектури та функціональної моделі веб-додатку автоматизації документообігу.
4. Реалізація веб-додатку з використанням серверної частини, бази даних та інтегрованого чат-боту.
5. Програмна реалізація функціоналу системи та проведення тестування розробленого рішення.

5.Перелік ілюстраційного матеріалу: *презентація*

6. Дата видачі завдання «20» лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/ п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Підбір технічної літератури.	21.02-02.03.25	Виконано
2.	Аналіз сучасних систем електронного документообігу та веб-технологій.	03.03-14.03.25	Виконано
3.	Дослідження можливостей використання веб-додатків та чат-ботів.	15.03-31.03.25	Виконано
4.	Проєктування архітектури та функціональної моделі системи документообігу.	01.04-24.04.25	Виконано
5.	Розробка веб-додатку та інтеграція чат-боту.	25.04-15.05.25	Виконано
6.	Проведення тестування та аналіз результатів роботи системи.	19.05-20.05.25	Виконано
7.	Оформлення бакалаврської роботи.	21.05-30.05.25	Виконано

Здобувач вищої освіти
(підпис)
Керівник кваліфікаційної роботи
(підпис)

Микола МИРОНЧУК
(ім'я, ПРІЗВИЩЕ)
Віктор САГАЙДАК
(ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступня бакалавр: 57 стор., 27 рис., 22 джерел.

Мета роботи – розробка веб-додатку для автоматизації документообігу з використанням сучасних веб-технологій, а також інтеграція чат-боту для спрощення взаємодії користувачів із системою.

Об'єкт дослідження – процес документообігу в інформаційних системах.

Предмет дослідження – методи та засоби розробки веб-додатків для автоматизації документообігу з використанням серверних технологій, баз даних та чат-ботів.

Короткий зміст роботи. У бакалаврській роботі проведено аналіз предметної області електронного документообігу та сучасних систем його автоматизації. Розглянуто особливості використання веб-технологій для створення інформаційних систем, а також можливості інтеграції чат-ботів у веб-додатки. Проаналізовано існуючі програмні рішення та визначено їх основні переваги і недоліки. У процесі виконання роботи було спроектовано архітектуру веб-додатку, визначено структуру бази даних та описано логіку взаємодії між компонентами системи. Реалізовано веб-додаток, який забезпечує створення, перегляд та обробку документів, а також контроль їх статусу. У практичній частині роботи розроблено серверну частину додатку з використанням мови програмування Python та фреймворку Flask. Для збереження даних використано реляційну базу даних SQLite.

КЛЮЧОВІ СЛОВА: ДОКУМЕНТООБІГ, ВЕБ-ДОДАТОК, АВТОМАТИЗАЦІЯ, ЧАТ-БОТ, FLASK, PYTHON, БАЗА ДАНИХ, SQLITE, ІНФОРМАЦІЙНА СИСТЕМА, ВЕБ-ТЕХНОЛОГІЇ.

ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ	12
1.1 Поняття та значення електронного документообігу	12
1.2 Аналіз процесів документообігу в організаціях	13
1.3 Огляд сучасних систем автоматизації документообігу	15
1.4 Використання чат-ботів у веб-додатках	17
1.5 Постановка задачі та вимоги до системи	19
1.6 Аналіз ролей користувачів у системах документообігу	20
1.7 Аналіз вимог до безпеки даних у системах документообігу	22
2 ПРОЄКТУВАННЯ ВЕБ-ДОДАТКУ ДЛЯ АВТОМАТИЗАЦІЇ ДОКУМЕНТООБІГУ	24
2.1 Формування функціональних та нефункціональних вимог	24
2.2 Розробка архітектури системи	25
2.3 Проєктування бази даних	27
2.4 Опис логіки роботи системи	29
2.5 Моделювання взаємодії користувача з системою	34
2.6 Проєктування системи ролей та доступів	35
2.7 Проєктування механізму обробки документів	38
2.8 Проєктування інтерфейсу користувача (UI/UX)	40
2.9 Проєктування механізму пошуку та фільтрації даних	43
3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБ-ДОДАТКУ	45
3.1 Вибір технологій та засобів розробки	45
3.2 Реалізація функціоналу веб-додатку	46
3.3 Розробка та інтеграція чат-боту	52
3.4 Реалізація системи авторизації та ролей доступу	54
3.5 Реалізація роботи з файлами документів	56
3.6 Реалізація пошуку та фільтрації документів	57
3.7 Аналіз продуктивності системи	59
3.8 Перспективи розвитку системи	60
3.9 Тестування системи	62
3.10 Оцінка ефективності розробленого рішення	63

ВИСНОВКИ	65
Список використаних джерел	67
ПЕРЕЛІК ДЕМОНСТРАЦІЙНИХ МАТЕРІАЛІВ (Презентація)	69

ВСТУП

У сучасних умовах стрімкого розвитку інформаційних технологій та цифровізації суспільства питання автоматизації внутрішніх процесів організацій набуває особливої актуальності [1]. Одним із ключових напрямів такого розвитку є вдосконалення систем документообігу, оскільки саме документи виступають основою управлінської діяльності, забезпечують фіксацію рішень та координацію дій між підрозділами.

Традиційні підходи до організації документообігу, що базуються на використанні паперових носіїв або частково автоматизованих систем, часто не відповідають сучасним вимогам ефективності. Вони характеризуються значними витратами часу на обробку документів, складністю контролю їх стану, а також підвищеним ризиком втрати або дублювання інформації. У зв'язку з цим виникає необхідність створення сучасних програмних рішень, які забезпечують централізоване зберігання даних, швидкий доступ до інформації та прозорість усіх етапів обробки документів.

Веб-технології відкривають широкі можливості для реалізації таких систем, оскільки дозволяють організувати доступ до функціоналу незалежно від місця перебування користувача та використовуваного пристрою. Водночас важливим напрямом розвитку інформаційних систем є впровадження чат-ботів, які забезпечують зручний інтерфейс взаємодії з користувачем [18]. Використання чат-ботів дає змогу автоматизувати виконання типових операцій, спростити доступ до інформації та підвищити швидкість обробки запитів, що особливо важливо в умовах зростання обсягів даних [19].

Аналіз сучасних наукових досліджень свідчить про значний інтерес до проблем автоматизації документообігу та розвитку веб-орієнтованих інформаційних систем [2]. Окрему увагу приділяють використанню інтелектуальних засобів взаємодії з користувачами, зокрема чат-ботів. Разом із тим питання ефективної інтеграції чат-ботів у системи документообігу потребує подальшого дослідження та практичної реалізації, що зумовлює доцільність виконання даної роботи.

Метою кваліфікаційної роботи є розробка веб-додатку для автоматизації документообігу з використанням чат-боту, який забезпечує підвищення ефективності обробки документів та спрощує взаємодію користувачів із системою. Досягнення поставленої мети передбачає дослідження особливостей організації документообігу, визначення вимог до програмної системи, проєктування її структури, а також реалізацію основного функціоналу та інтеграцію чат-боту.

Об'єктом дослідження є процес документообігу в організаціях, що охоплює створення, обробку, зберігання та контроль документів. Предметом дослідження виступають методи та засоби автоматизації документообігу з використанням сучасних веб-технологій і чат-ботів.

У процесі виконання роботи застосовано методи аналізу наукових джерел, узагальнення теоретичних положень, проєктування інформаційних систем та програмної реалізації. Використання зазначених методів дозволило обґрунтувати вибір архітектури системи та забезпечити її функціональність відповідно до поставлених завдань.

Наукова новизна отриманих результатів полягає у формуванні підходу до побудови веб-додатку для автоматизації документообігу з інтегрованим чат-ботом, що забезпечує більш ефективну взаємодію користувачів із системою. Практична значущість роботи полягає у можливості використання розробленого програмного продукту в діяльності організацій для оптимізації процесів обробки документів та підвищення продуктивності праці.

Кваліфікаційна робота має традиційну структуру і складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

Апробація результатів та публікації. Основні положення і результати бакалаврської роботи доповідались на VII Міжнародній науково-технічній конференції «Сучасний стан та перспективи розвитку IoT» з тезами на тему «Веб-додаток для автоматизації документообігу з використанням чат-боту», а також на Всеукраїнській науково-технічній конференції «Технології цифрового розвитку: програмні системи та децентралізація» з тезами на тему «Система

інтелектуальної маршрутизації цифрових документів на основі чат-інтерфейсів» що проходили на базі Державного університету інформаційно-комунікаційних технологій.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

1.1 Поняття та значення електронного документообігу

У сучасних умовах розвитку інформаційного суспільства документообіг є невід’ємною складовою діяльності будь-якої організації. Він забезпечує фіксацію управлінських рішень, передачу інформації між підрозділами та контроль виконання поставлених завдань. З переходом до цифрових технологій традиційні паперові процеси поступово замінюються електронними, що зумовлює формування нового підходу до організації роботи з документами.

Електронний документообіг являє собою сукупність процесів створення, обробки, передачі, зберігання та використання документів у цифровій формі із застосуванням інформаційно-комунікаційних технологій (рис. 1.1). На відміну від паперових систем, електронний документообіг забезпечує значно вищу швидкість обробки інформації, зручність доступу до документів та можливість автоматизації багатьох операцій [20].

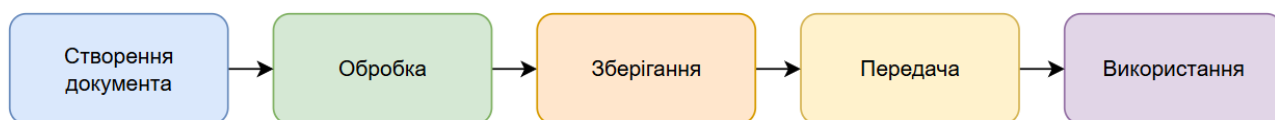


Рис. 1.1 — Загальна схема електронного документообігу

Основною особливістю електронного документообігу є використання електронних документів, які мають юридичну силу за умови дотримання встановлених вимог, зокрема застосування електронного підпису. Це дозволяє організаціям повністю або частково відмовитися від паперових носіїв, що, у свою чергу, сприяє зменшенню витрат на зберігання документів та оптимізації внутрішніх процесів.

Значення електронного документообігу полягає насамперед у підвищенні ефективності управлінської діяльності. Завдяки автоматизації обробки документів скорочується час на їх створення та погодження, підвищується точність виконання

операцій, а також забезпечується прозорість усіх етапів роботи з документами [7]. Керівництво організації отримує можливість оперативно контролювати стан виконання завдань і приймати обґрунтовані управлінські рішення.

Крім того, електронний документообіг сприяє покращенню взаємодії між працівниками та підрозділами. Завдяки централізованому зберіганню даних забезпечується швидкий доступ до необхідної інформації, що зменшує кількість помилок та дублювання даних. Це особливо важливо для організацій із великою кількістю документів, де ефективність обробки інформації безпосередньо впливає на результати діяльності.

Важливим аспектом є також підвищення рівня безпеки інформації. Сучасні системи електронного документообігу передбачають використання механізмів авторизації, контролю доступу та шифрування даних, що дозволяє захистити документи від несанкціонованого доступу та втрати [12]. У порівнянні з паперовими архівами електронні системи забезпечують більш надійне зберігання інформації та можливість її резервного копіювання.

У контексті цифрової трансформації бізнесу електронний документообіг виступає важливим інструментом підвищення конкурентоспроможності організацій. Його впровадження дозволяє оптимізувати внутрішні процеси, скоротити витрати та підвищити якість обслуговування клієнтів. Саме тому розробка та впровадження сучасних систем електронного документообігу є актуальним завданням для більшості підприємств і установ.

Таким чином, електронний документообіг є ключовим елементом сучасних інформаційних систем, який забезпечує ефективну організацію роботи з документами та створює передумови для подальшої автоматизації управлінських процесів.

1.2 Аналіз процесів документообігу в організаціях

Процеси документообігу є основою функціонування будь-якої організації, оскільки саме через документи здійснюється передача інформації, прийняття

управлінських рішень та контроль їх виконання. Незалежно від сфери діяльності, документообіг охоплює повний життєвий цикл документа — від моменту його створення до завершення використання або архівування.

У загальному вигляді процес документообігу включає кілька послідовних етапів(рис. 1.2), кожен з яких має власне функціональне призначення. На початковому етапі відбувається створення або отримання документа, після чого він підлягає реєстрації в системі. Далі документ передається на розгляд або обробку відповідальним особам, де можуть виконуватися такі дії, як погодження, редагування або затвердження. Після завершення обробки документ або передається для подальшого використання, або зберігається в архіві [13].

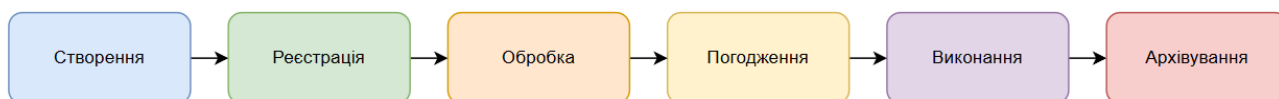


Рис. 1.2 — Життєвий цикл документа

У традиційних організаціях значна частина цих процесів виконується вручну або з мінімальним рівнем автоматизації. Це призводить до затримок у обробці документів, складності відстеження їхнього стану та підвищення ймовірності помилок. Крім того, у разі великого обсягу документів виникають труднощі з їх пошуком і контролем виконання, що негативно впливає на загальну ефективність діяльності організації.

Особливу увагу слід приділити проблемам, що виникають у процесі документообігу. Однією з основних є відсутність централізованої системи зберігання інформації, що призводить до дублювання документів та втрати актуальних даних. Іншою важливою проблемою є недостатній контроль за виконанням документів, унаслідок чого окремі завдання можуть залишатися невиконаними або виконуватися із запізненням [9]. Також варто відзначити значні витрати часу на погодження документів, особливо у випадках, коли вони повинні проходити через декілька рівнів управління.

Перехід до електронного документообігу дозволяє вирішити більшість зазначених проблем шляхом автоматизації ключових процесів. Використання інформаційних систем забезпечує можливість швидкої реєстрації документів, автоматичного розподілу між відповідальними особами та контролю виконання в режимі реального часу. Крім того, електронні системи дозволяють зберігати всі документи в єдиному інформаційному середовищі, що значно спрощує їх пошук та використання.

Важливим аспектом є також можливість інтеграції документообігу з іншими бізнес-процесами організації. Це дозволяє створити єдину інформаційну систему, у межах якої документи стають частиною загального процесу управління. У такому випадку документообіг перестає бути окремою функцією і перетворюється на невід'ємний елемент інформаційної інфраструктури організації.

Таким чином, аналіз процесів документообігу свідчить про необхідність їх автоматизації та оптимізації. Використання сучасних інформаційних технологій, зокрема веб-додатків і чат-ботів, дозволяє значно підвищити ефективність обробки документів, зменшити витрати часу та забезпечити прозорість усіх етапів роботи з інформацією.

1.3 Огляд сучасних систем автоматизації документообігу

Сучасний етап розвитку інформаційних технологій характеризується широким впровадженням систем автоматизації документообігу в діяльність організацій різного масштабу. Такі системи дозволяють оптимізувати процеси обробки документів, підвищити ефективність управління інформацією та забезпечити контроль за виконанням завдань. У зв'язку з цим ринок програмних рішень у сфері електронного документообігу є достатньо розвиненим і представлений як комерційними, так і відкритими продуктами.

Серед найбільш поширених систем можна виділити рішення корпоративного рівня, які орієнтовані на великі підприємства та державні установи. До таких систем належать, зокрема, Microsoft SharePoint, M-Files та

DocuWare [21]. Вони забезпечують комплексний підхід до управління документами, включаючи автоматизацію бізнес-процесів, контроль доступу, інтеграцію з іншими інформаційними системами та використання хмарних технологій. На рисунку 1.3 наведено узагальнену класифікацію сучасних систем документообігу залежно від способу їх реалізації та доступу до функціоналу.

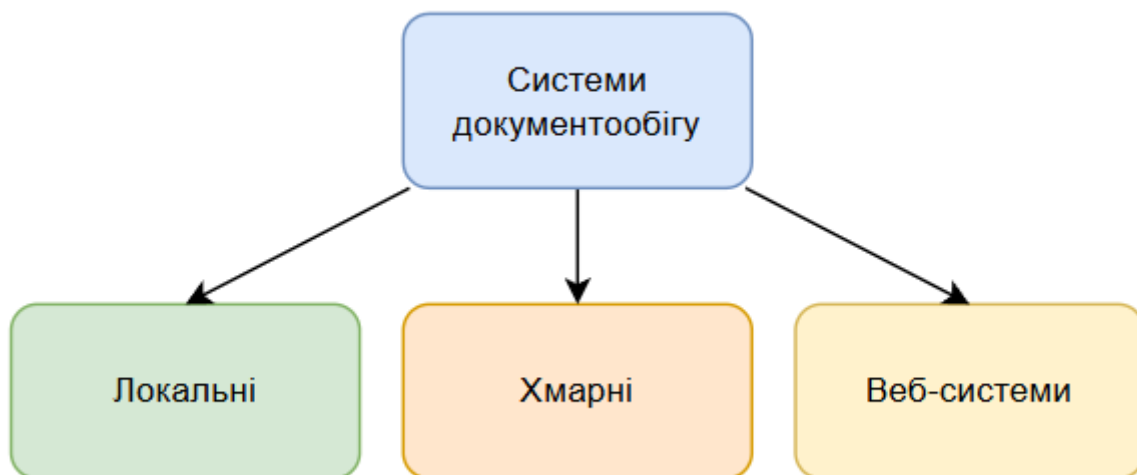


Рис. 1.3 — Класифікація систем документообігу

Значна частина сучасних систем документообігу реалізується у вигляді веб-додатків або хмарних сервісів, що забезпечує доступ до них через браузер без необхідності встановлення додаткового програмного забезпечення [18]. Такий підхід дозволяє користувачам працювати з документами незалежно від місця їх перебування, що особливо важливо в умовах віддаленої роботи та розподілених команд.

Важливою тенденцією є інтеграція систем документообігу з іншими програмними продуктами, такими як системи управління підприємством, електронна пошта та засоби комунікації. Це дозволяє створити єдине інформаційне середовище, у якому документи стають частиною загального процесу управління організацією [1]. У результаті зменшується кількість ручних операцій, підвищується точність обробки даних і скорочується час виконання завдань.

Окрему увагу в сучасних системах приділяють інтерфейсу користувача та зручності роботи з системою. Інтуїтивно зрозумілий інтерфейс дозволяє зменшити час на навчання персоналу та підвищити ефективність використання програмного забезпечення. У цьому контексті особливого значення набувають нові підходи до взаємодії з користувачем, зокрема використання чат-ботів та голосових помічників.

Незважаючи на значні переваги існуючих систем, вони мають і певні недоліки. Більшість корпоративних рішень є складними у впровадженні, потребують значних фінансових витрат та тривалого налаштування. Крім того, такі системи часто перевантажені функціоналом, який не завжди необхідний для невеликих організацій. Це створює передумови для розробки більш простих і гнучких рішень, орієнтованих на конкретні потреби користувачів [2].

Таким чином, аналіз сучасних систем автоматизації документообігу показує, що існує широкий спектр програмних рішень, які дозволяють ефективно організувати роботу з документами. Водночас актуальним залишається питання створення доступних, простих у використанні та інтегрованих систем, які поєднують можливості веб-додатків і чат-ботів, що і визначає напрям подальшого дослідження.

1.4 Використання чат-ботів у веб-додатках

Сучасний розвиток інформаційних технологій сприяє появі нових підходів до організації взаємодії користувача з програмними системами. Одним із таких підходів є використання чат-ботів, які дозволяють реалізувати інтуїтивно зрозумілий та зручний спосіб комунікації на основі текстового або голосового введення. У контексті веб-додатків чат-боти виступають як інструмент автоматизації взаємодії між користувачем і системою.

Чат-бот являє собою програмний компонент, здатний імітувати діалог із користувачем та виконувати певні дії на основі отриманих запитів. У веб-додатках такі боти можуть бути інтегровані безпосередньо в інтерфейс системи, що

забезпечує швидкий доступ до їх функціоналу без необхідності використання сторонніх сервісів. Це дозволяє значно підвищити зручність користування системою та скоротити час на виконання типових операцій [18].

Однією з основних переваг чат-ботів є можливість автоматизації рутинних процесів. Замість того щоб переходити між різними розділами веб-додатку, користувач може виконати необхідну дію за допомогою текстової команди. Наприклад, у системах документообігу чат-бот може надати інформацію про кількість документів, їх статус або допомогти створити новий документ. Це значно спрощує взаємодію з системою, особливо для користувачів, які не мають достатнього досвіду роботи з подібними програмними продуктами [19].

Важливою характеристикою чат-ботів є їх гнучкість та можливість адаптації до потреб користувачів. Прості боти працюють на основі заздалегідь визначених правил і ключових слів, тоді як більш складні системи можуть використовувати технології обробки природної мови [3]. У межах веб-додатків для автоматизації документообігу доцільним є використання правил-орієнтованих чат-ботів, які забезпечують швидку реалізацію та достатній рівень функціональності.

Інтеграція чат-ботів у веб-додатки також сприяє підвищенню ефективності роботи системи в цілому. Завдяки автоматичній обробці запитів зменшується навантаження на інтерфейс та скорочується кількість дій, які користувач повинен виконувати вручну. Крім того, чат-бот може виступати як допоміжний інструмент, який надає підказки щодо використання системи та допомагає швидко знайти необхідну інформацію.

Разом із тим використання чат-ботів має і певні обмеження. Зокрема, їх функціональність залежить від якості реалізації алгоритмів обробки запитів, а також від повноти набору команд, які вони підтримують. У разі складних або нестандартних запитів чат-бот може не забезпечити коректну відповідь, що потребує подальшого вдосконалення таких систем.

Незважаючи на зазначені обмеження, чат-боти є перспективним напрямом розвитку веб-додатків, оскільки дозволяють зробити взаємодію з системою більш природною та ефективною. У поєднанні з системами електронного

документообігу вони створюють нові можливості для автоматизації процесів та підвищення продуктивності роботи користувачів.

Таким чином, використання чат-ботів у веб-додатках є доцільним і обґрунтованим підходом, що дозволяє підвищити зручність роботи з системою та забезпечити швидкий доступ до її функціональних можливостей.

1.5 Постановка задачі та вимоги до системи

Проведений аналіз предметної області документообігу, а також огляд сучасних систем автоматизації показали, що, незважаючи на значну кількість існуючих рішень, проблема ефективної організації роботи з документами залишається актуальною. Особливо це стосується невеликих і середніх організацій, для яких складні корпоративні системи є надто дорогими або важкими у впровадженні та використанні.

У більшості випадків існуючі програмні продукти характеризуються надлишковим функціоналом, складним інтерфейсом та високими вимогами до ресурсів. Це ускладнює їх застосування в умовах, коли необхідно забезпечити швидке та зручне виконання базових операцій з документами. Крім того, не всі системи підтримують сучасні підходи до взаємодії з користувачем, зокрема використання чат-ботів як інструменту спрощення роботи з інформацією [10].

У зв'язку з цим виникає необхідність розробки веб-додатку, який поєднує простоту використання, доступність та достатній рівень функціональності для автоматизації основних процесів документообігу. Такий додаток повинен забезпечувати можливість створення документів, їх обробки, зміни статусу та контролю виконання, а також надавати користувачу зручні засоби взаємодії із системою.

Особливістю розроблюваної системи є інтеграція чат-боту, який виступає як додатковий інтерфейс взаємодії. Завдяки цьому користувач отримує можливість виконувати типові операції та отримувати інформацію у більш зручній формі, без

необхідності переходу між різними елементами інтерфейсу. Такий підхід дозволяє підвищити ефективність роботи та зменшити час на виконання стандартних дій.

Розроблювана система повинна бути реалізована у вигляді веб-додатку, що забезпечує доступ до її функціоналу через браузер. Це дозволяє використовувати систему без встановлення додаткового програмного забезпечення та забезпечує можливість роботи з нею з різних пристроїв [21]. Крім того, система повинна підтримувати збереження даних у базі даних, що забезпечує їх цілісність та доступність.

Функціональні можливості системи мають включати базові операції з документами, зокрема створення нових записів, перегляд інформації, зміну статусу та зберігання даних. Важливим аспектом є також забезпечення авторизації користувачів, що дозволяє контролювати доступ до системи та розмежовувати права користувачів.

Окрім функціональних характеристик, система повинна відповідати ряду нефункціональних вимог, серед яких особливе значення мають зручність використання, швидкість роботи, надійність та безпека. Інтерфейс системи має бути інтуїтивно зрозумілим, що дозволяє користувачам швидко освоїти її функціонал. Швидкість обробки запитів повинна забезпечувати комфортну роботу без затримок, а механізми захисту даних мають запобігати несанкціонованому доступу.

Таким чином, постановка задачі полягає у розробці веб-додатку для автоматизації документообігу з інтегрованим чат-ботом, який забезпечує ефективну обробку документів, зручну взаємодію з користувачем та відповідає сучасним вимогам до інформаційних систем. Отримані результати аналізу слугують основою для подальшого проєктування та реалізації системи, що розглядається у наступних розділах роботи.

1.6 Аналіз ролей користувачів у системах документообігу

У сучасних системах електронного документообігу важливим аспектом є організація доступу користувачів до функціоналу системи. Це досягається за допомогою впровадження ролей, які визначають права та обмеження кожного користувача. Використання ролей дозволяє забезпечити контроль доступу до інформації, підвищити рівень безпеки та оптимізувати роботу з документами.

Ролі користувачів у системах документообігу формуються залежно від структури організації та бізнес-процесів, що в ній реалізуються. Кожна роль відповідає певному рівню доступу до даних і функцій системи. Такий підхід дозволяє розмежувати відповідальність між користувачами та уникнути несанкціонованих дій [6].

У більшості систем можна виділити кілька основних категорій ролей. Адміністратор має повний доступ до системи, включаючи керування користувачами, налаштування параметрів роботи та контроль усіх документів. Така роль передбачає найвищий рівень відповідальності, оскільки саме адміністратор забезпечує коректне функціонування системи.

Користувачі з управлінськими функціями, яких часто називають менеджерами або модераторами, мають розширені права у порівнянні зі звичайними користувачами. Вони можуть переглядати документи, змінювати їх статус, здійснювати погодження або відхилення. Таким чином, вони беруть участь у процесі обробки документів та забезпечують виконання бізнес-процесів.

Звичайні користувачі мають обмежений доступ до системи. Вони можуть створювати нові документи, переглядати їх та, у деяких випадках, завантажувати або скачувати файли. При цьому їх можливості редагування або зміни статусу документів, як правило, обмежені, що дозволяє уникнути помилок та несанкціонованих змін.

Окремо варто виділити роль користувача з мінімальним рівнем доступу, який не має можливості взаємодіяти з документами. Така роль може використовуватися для нових або неавторизованих користувачів, доступ яких ще не підтверджено. Це дозволяє забезпечити додатковий рівень безпеки та контроль над доступом до системи.

Використання ролей також тісно пов'язане із забезпеченням інформаційної безпеки. Розмежування доступу дозволяє обмежити коло осіб, які можуть працювати з конфіденційною інформацією, а також зменшує ризик випадкового або навмисного пошкодження даних [12]. Крім того, це спрощує аудит дій користувачів, оскільки кожна дія може бути прив'язана до конкретної ролі.

Таким чином, система ролей є невід'ємною частиною будь-якої сучасної системи документообігу. Вона забезпечує ефективне управління доступом, підвищує безпеку та дозволяє організувати роботу користувачів відповідно до їхніх функцій. Врахування цих аспектів є важливим при проєктуванні власної системи автоматизації документообігу.

1.7 Аналіз вимог до безпеки даних у системах документообігу

У системах електронного документообігу питання безпеки даних є одним із ключових, оскільки такі системи працюють із великою кількістю важливої, а інколи й конфіденційної інформації. Документи можуть містити персональні дані, фінансову інформацію, службову переписку та інші відомості, доступ до яких повинен бути строго обмежений [12]. Тому забезпечення захисту інформації є обов'язковою вимогою при розробці та впровадженні подібних систем.

Однією з основних вимог до безпеки є автентифікація користувачів. Вона дозволяє переконатися, що доступ до системи отримує саме той користувач, який має відповідні права. Зазвичай це реалізується за допомогою введення логіну та пароля, однак у більш складних системах можуть використовуватися додаткові механізми, такі як двофакторна автентифікація [2].

Не менш важливою є авторизація, яка визначає, які саме дії може виконувати користувач після входу в систему. Саме тут використовується механізм ролей, що дозволяє обмежити доступ до певних функцій та даних. Наприклад, звичайний користувач може лише переглядати документи, тоді як менеджер має право змінювати їх статус, а адміністратор — повністю керувати системою.

Важливим аспектом безпеки є захист даних під час їх зберігання. Інформація в базі даних повинна бути захищена від несанкціонованого доступу, а критично важливі дані, такі як паролі користувачів, повинні зберігатися у зашифрованому вигляді. Це дозволяє мінімізувати ризики у випадку витоку даних [1].

Окрему увагу слід приділити захисту даних під час їх передачі. Взаємодія між клієнтською та серверною частинами системи повинна здійснюватися через захищені протоколи, що запобігають перехопленню інформації сторонніми особами. Це особливо важливо для веб-додатків, які працюють у відкритих мережах.

У системах документообігу також важливо забезпечити контроль доступу до файлів. Оскільки документи можуть містити вкладення у вигляді файлів, необхідно обмежити можливість їх перегляду та завантаження лише для авторизованих користувачів. Крім того, слід контролювати типи файлів, які можуть бути завантажені, щоб запобігти розповсюдженню шкідливого програмного забезпечення.

Ще однією важливою вимогою є забезпечення цілісності даних. Система повинна гарантувати, що документи не можуть бути змінені або видалені без відповідних прав доступу. Для цього застосовуються механізми перевірки прав користувачів, а також фіксація змін у системі.

Крім того, важливим елементом безпеки є можливість відновлення даних у разі їх втрати або пошкодження. Для цього використовуються резервні копії, які дозволяють відновити інформацію та забезпечити безперервність роботи системи [4].

Таким чином, забезпечення безпеки даних у системах документообігу є комплексною задачею, яка включає автентифікацію, авторизацію, захист даних при зберіганні та передачі, контроль доступу до файлів і забезпечення цілісності інформації. Дотримання цих вимог дозволяє створити надійну систему, яка відповідає сучасним стандартам інформаційної безпеки.

2 ПРОЄКТУВАННЯ ВЕБ-ДОДАТКУ ДЛЯ АВТОМАТИЗАЦІЇ ДОКУМЕНТООБІГУ

2.1 Формування функціональних та нефункціональних вимог

На основі проведеного аналізу предметної області та існуючих рішень формується перелік вимог до розроблюваної системи автоматизації документообігу. Визначення вимог є важливим етапом проєктування, оскільки саме вони задають напрям подальшої розробки та визначають функціональні можливості системи.

Функціональні вимоги описують основні можливості системи, які забезпечують виконання поставлених задач. Розроблюваний веб-додаток повинен надавати користувачам можливість працювати з документами у зручному та зрозумілому середовищі. Система повинна забезпечувати створення нових документів із зазначенням їх основних характеристик, таких як назва, тип та опис. Користувач повинен мати змогу переглядати список документів, отримувати інформацію про їх стан, а також змінювати статус залежно від етапу обробки [12].

Важливим елементом функціональності є забезпечення доступу до системи через механізм авторизації. Це дозволяє ідентифікувати користувачів та контролювати їх дії в межах системи. Крім того, система повинна забезпечувати збереження всієї інформації у базі даних, що гарантує цілісність та доступність даних.

Особливу роль у розроблюваній системі відіграє чат-бот, який виступає як додатковий інтерфейс взаємодії з користувачем. Він повинен забезпечувати обробку текстових запитів, надавати інформацію про документи, їх кількість та статуси, а також допомагати користувачу орієнтуватися у функціоналі системи. Завдяки цьому значно спрощується виконання типових операцій та підвищується зручність використання додатку [18].

Окрім функціональних можливостей, система повинна відповідати нефункціональним вимогам, які визначають якість її роботи. Однією з основних характеристик є зручність використання, що передбачає наявність інтуїтивно

зрозумілого інтерфейсу та логічної структури навігації. Користувач повинен мати змогу швидко освоїти систему без необхідності тривалого навчання.

Не менш важливою є продуктивність системи, яка повинна забезпечувати швидку обробку запитів та відображення інформації без помітних затримок. Це досягається за рахунок ефективної організації серверної частини та оптимізації взаємодії з базою даних.

Система також повинна забезпечувати надійність зберігання даних, що передбачає їх захист від втрати або пошкодження. Використання механізмів резервного копіювання та контроль доступу дозволяє мінімізувати ризики несанкціонованого використання інформації.

Окрему увагу слід приділити масштабованості системи, що дає змогу розширювати її функціональні можливості у майбутньому без суттєвих змін у структурі. Це особливо важливо з урахуванням можливого збільшення кількості користувачів та обсягу оброблюваних даних.

Таким чином, сформовані функціональні та нефункціональні вимоги визначають основні характеристики розроблюваного веб-додатку та створюють основу для його подальшого проєктування і реалізації.

2.2 Розробка архітектури системи

Розробка архітектури інформаційної системи є ключовим етапом проєктування, оскільки саме вона визначає структуру програмного забезпечення, взаємозв'язки між його компонентами та принципи обробки даних. Для системи автоматизації документообігу з використанням чат-боту було обрано архітектурний підхід, який забезпечує простоту реалізації, гнучкість та можливість подальшого розширення функціоналу.

Основою розроблюваного рішення є клієнт-серверна архітектура, у межах якої відбувається розподіл функцій між користувацькою та серверною частинами. Користувач взаємодіє із системою через веб-інтерфейс, який відображається у браузері, тоді як основна логіка обробки даних реалізується на сервері [1]. Такий

підхід дозволяє централізувати обробку інформації та забезпечити доступ до системи з різних пристроїв.

Клієнтська частина системи відповідає за відображення інформації та взаємодію з користувачем. Вона реалізована у вигляді веб-сторінок, які забезпечують можливість створення, перегляду та редагування документів. Окрім цього, у клієнтський інтерфейс інтегровано модуль чат-боту, який дозволяє користувачу взаємодіяти із системою за допомогою текстових повідомлень.

Серверна частина системи виконує основну обробку запитів та реалізує бізнес-логіку додатку. Вона забезпечує прийом даних від клієнта, їх обробку, взаємодію з базою даних та формування відповіді. У межах розроблюваного веб-додатку сервер реалізовано з використанням фреймворку Flask, який дозволяє створювати легкі та гнучкі веб-застосунки [16].

Важливим компонентом системи є база даних, яка використовується для збереження інформації про користувачів та документи. Застосування реляційної бази даних забезпечує структуроване зберігання даних, їх цілісність та можливість швидкого доступу до необхідної інформації. Взаємодія між сервером і базою даних здійснюється за допомогою відповідних запитів, що дозволяє ефективно керувати інформацією.

Окреме місце в архітектурі займає чат-бот, який інтегрований у серверну частину системи. Він обробляє текстові запити користувача, аналізує їх та формує відповідь відповідно до закладеної логіки. Завдяки цьому чат-бот виступає як проміжний рівень між користувачем і системою, спрощуючи доступ до функціональних можливостей додатку.

Обрана архітектура дозволяє забезпечити модульність системи, що дає змогу легко змінювати або розширювати окремі компоненти без необхідності повної перебудови системи. Це створює передумови для подальшого розвитку програмного продукту, зокрема додавання нових функцій або інтеграції з іншими сервісами.

Таким чином, розроблена архітектура системи відповідає сучасним вимогам до веб-додатків, забезпечує ефективну обробку даних, зручну взаємодію з користувачем та можливість подальшого масштабування.

2.3 Проєктування бази даних

Проєктування бази даних є важливим етапом створення веб-додатку для автоматизації документообігу, оскільки саме база даних забезпечує збереження, обробку та подальше використання інформації в системі. Для розроблюваного програмного продукту база даних повинна зберігати дані про користувачів, документи, їх статуси та основні характеристики.

У межах створеного веб-додатку використовується реляційна модель даних, оскільки вона є зручною для зберігання структурованої інформації та дозволяє встановлювати зв'язки між окремими сутностями. Основними сутностями системи є користувач та документ. Користувач виконує дії в системі, створює документи та змінює їх стан, а документ є основним об'єктом обробки в межах документообігу [5].

Для збереження інформації про користувачів передбачено таблицю users(рис. 2.3.1). Вона містить дані, необхідні для реєстрації та авторизації в системі. Зокрема, у цій таблиці зберігається унікальний ідентифікатор користувача, логін, хешований пароль та роль користувача. Використання хешованого пароля підвищує рівень безпеки, оскільки реальний пароль не зберігається у відкритому вигляді.

id	username	password_hash	role
1	антон	bcrypt:32768:8:1\$V5H8MDpArP0RydUt\$07791158796d7b9df4d041b4fed...	admin
2	Евген	bcrypt:32768:8:1\$ziYZse12APtaob3e\$eadb0b9bb8b761446ca7ca8fbcd...	manager
3	Аня	bcrypt:32768:8:1\$v3KMNrJw6E6xPiUW\$f2b28b5404ef99a969da1f7f028...	user
4	Оля	bcrypt:32768:8:1\$Jq6PjYlIcsLtMA6o\$15d1b2e0cb338d21d64923127ad...	manager
5	Вадим	bcrypt:32768:8:1\$LbT9e2G4m7606Yzd\$cb80cf8de20186702dd6cf78b36...	user
6	Віталій	bcrypt:32768:8:1\$cnfit41Jv1voWRaD\$bbebb04010587cb88cc4a4d6cb2...	quest

Рис. 2.3.1 – Таблиця users

Для збереження інформації про документи використовується таблиця documents(рис. 2.3.2). Вона містить назву документа, його тип, опис, поточний статус, дату створення, дату оновлення та ідентифікатор користувача, який створив документ. Завдяки цьому система може відображати список документів, контролювати їх стан та зберігати історично важливі дані про час створення й останнього оновлення.

id	title	doc_type	description	status	
1	Звіт за 4 квартал	Звіт	Звіт по зарплаті	Новий	
2	Заява на роботу Поповійчук	Заява	Заява від Поповійчук	В обробці	
3	Наказ 1	Наказ	Наказ про відпустку	Новий	
4	Акт виконаних робіт	Акт	Акт виконаних робіт	Новий	

created_by	file_name	original_file_name	created_at	updated_at	
2	20260426112547_2_1.pdf	1.pdf	2026-04-26 11:25:47	2026-04-26 11:25:47	
2	20260426112626_2_ilo.docx	ilo.docx	2026-04-26 11:26:26	2026-04-26 11:26:36	
3	20260426112741_3_docx	docx	2026-04-26 11:27:41	2026-04-26 11:27:41	
3	20260426112809_3_6.docx	6.docx	2026-04-26 11:28:09	2026-04-26 11:28:09	

Рис. 2.3.2 – Таблиця documents

Зв'язок між таблицями реалізується через поле created_by, яке вказує на користувача, що створив відповідний документ. Такий підхід дозволяє визначити автора кожного документа та забезпечити логічну цілісність даних. У разі подальшого розвитку системи ця структура може бути доповнена новими таблицями, наприклад для зберігання історії змін, коментарів або вкладених файлів [22].

Обрана структура бази даних є достатньою для реалізації основного функціоналу веб-додатку. Вона забезпечує збереження користувачів, документів та їх статусів, а також дозволяє швидко отримувати необхідну інформацію для відображення у веб-інтерфейсі та формування відповідей чат-ботом.

Таким чином, спроектована база даних відповідає функціональним вимогам системи та забезпечує надійну основу для роботи веб-додатку автоматизації документообігу.

2.4 Опис логіки роботи системи

Логіка роботи розроблюваної системи автоматизації документообігу визначає порядок взаємодії між користувачем, веб-інтерфейсом, серверною частиною та базою даних. Вона охоплює всі основні сценарії використання системи та забезпечує виконання функціональних можливостей, визначених на попередніх етапах проєктування [10].

Робота системи починається з авторизації користувача(рис. 2.4.1, рис. 2.4.2, рис. 2.4.3).

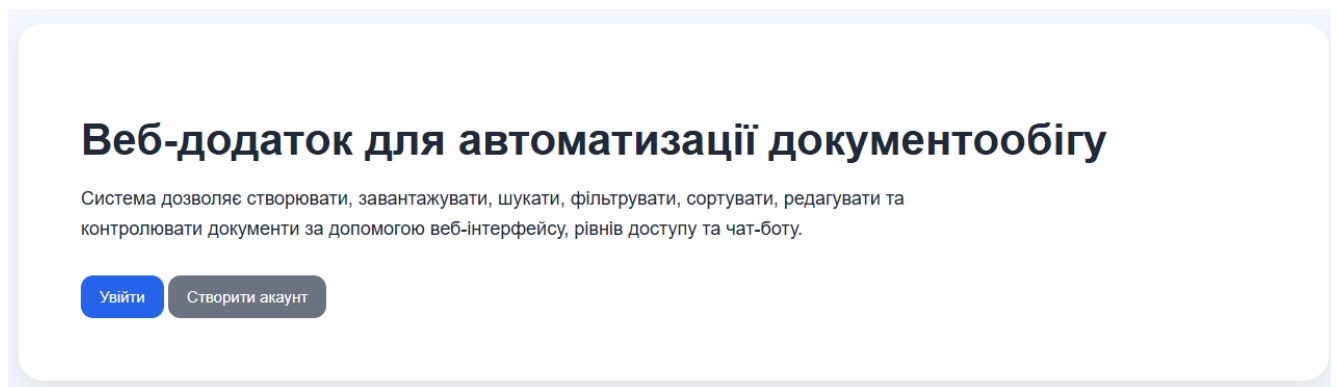


Рис. 2.4.1 – Перша сторінка веб-додатку

The registration form is titled "Реєстрація" in bold black text. It contains two input fields: "Логін" (Login) and "Пароль" (Password). Below the password field is a blue button with the text "Зареєструватися" (Register). At the bottom, there is a note in gray text: "Після реєстрації акаунт отримує роль guest. Для доступу передайте username керівництву." (After registration, the account receives the role guest. To access, provide the username to management.)

Рис. 2.4.2 – Реєстрація

The login form is titled "Вхід" (Login) in bold black text. It contains two input fields: "Логін" (Login) and "Пароль" (Password). Below the password field is a blue button with the text "Увійти" (Login).

Рис. 2.4.3 – Вхід

Після введення логіну та пароля відповідні дані передаються на сервер, де здійснюється їх перевірка(рис. 2.4.4). У разі успішної автентифікації користувач отримує доступ до функціоналу системи, зокрема до панелі керування

документами. Такий підхід дозволяє забезпечити контроль доступу та захист даних від несанкціонованого використання.

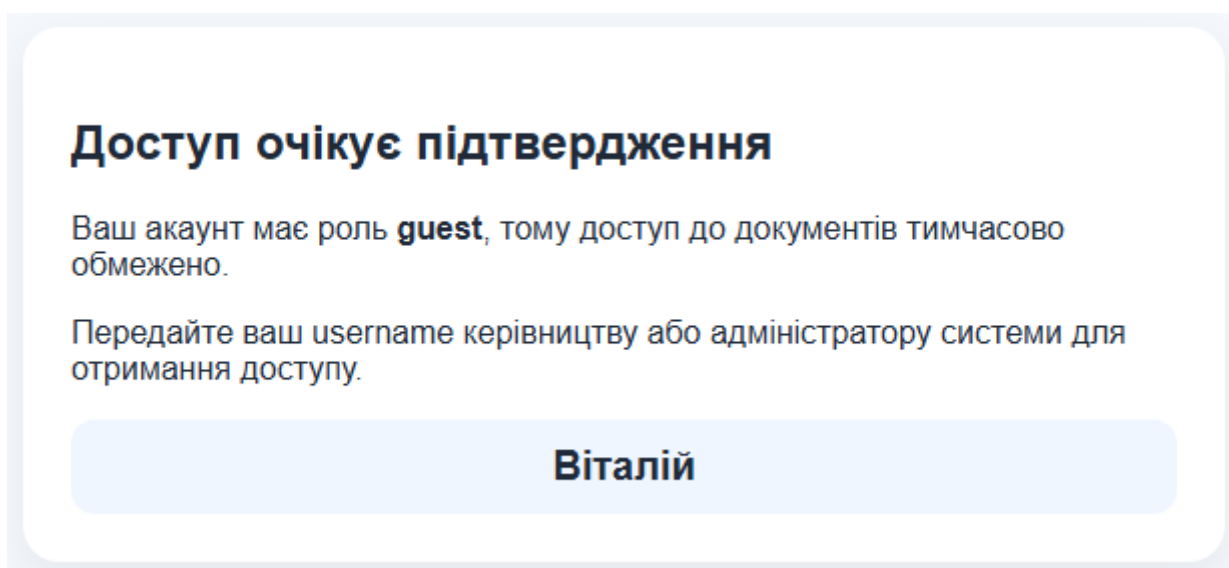


Рис. 2.4.4 – Підтвердження

Після входу в систему користувач має можливість переглядати список документів, що зберігаються в базі даних (рис. 2.4.5). Дані завантажуються з сервера у вигляді структурованої інформації та відображаються у веб-інтерфейсі. Для кожного документа доступні основні характеристики, включаючи назву, тип, опис, статус та дату створення. Це дозволяє користувачу швидко отримати повну інформацію про стан документообігу.

ID	Назва	Тип	Опис	Файл	Автор	Статус	Дата	Дії
4	Акт виконаних робіт	Акт	Акт виконаних робіт	Завантажити	Аня	Новий	2026-04-26 11:28:09	Перегляд
3	Наказ 1	Наказ	Наказ про відпустку	Завантажити	Аня	Новий	2026-04-26 11:27:41	Перегляд
2	Заява на роботу Поповійчук	Заява	Заява від Поповійчук	Завантажити	Евген	В обробці	2026-04-26 11:26:26	Перегляд
1	Звіт за 4 квартал	Звіт	Звіт по зарплаті	Завантажити	Евген	Новий	2026-04-26 11:25:47	Перегляд

Рис. 2.4.5 – Список документів

Одним із ключових сценаріїв роботи є створення нового документа. Користувач вводить необхідні дані через форму у веб-інтерфейсі (рис. 2.4.6), після

чого вони передаються на сервер для обробки. Сервер перевіряє коректність введених даних, формує запис та зберігає його у базі даних. Після цього новий документ стає доступним у загальному списку.

Іншим важливим елементом логіки роботи є зміна статусу документа(рис. 2.4.7). Користувач може обрати відповідний статус залежно від етапу обробки, наприклад, перевести документ у стан обробки, погодження або завершення. Після виконання цієї дії інформація оновлюється у базі даних, що дозволяє відслідковувати поточний стан кожного документа.

Особливу роль у системі відіграє чат-бот, який інтегрований у веб-інтерфейс і забезпечує альтернативний спосіб взаємодії з системою(рис. 2.4.8). Користувач може вводити текстові запити, які передаються на сервер у вигляді повідомлень. Серверна частина аналізує зміст запиту та формує відповідь відповідно до заданої логіки. Наприклад, користувач може отримати інформацію про кількість документів, їх статус або рекомендації щодо виконання певних дій.

Створення документа

Назва документа

Наприклад: Звіт за 4 квартал

Тип документа

Заява

Опис

Короткий опис документа

Файл документа

Вибрати файл Файл не вибрано

Дозволені формати: PDF, DOC, DOCX, XLS, XLSX, TXT, PNG, JPG.

Створити

Рис. 2.4.6 – Створення документа

Дії

Новий

Новий

В обробці

Погоджено

Відхилено

Архів

Рис. 2.4.7 – Зміна статусу документа

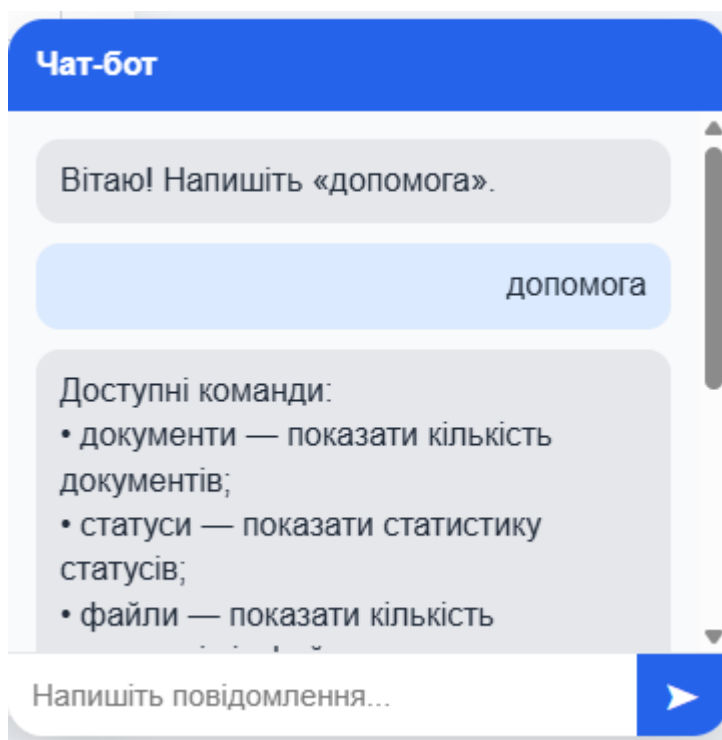


Рис. 2.4.8 – Чат-бот

Обробка запитів чат-ботом здійснюється на основі визначених правил, що дозволяє швидко реагувати на стандартні команди. Такий підхід є достатнім для реалізації базового функціоналу та не потребує використання складних алгоритмів обробки природної мови. Водночас система може бути розширена у майбутньому шляхом додавання більш складних механізмів аналізу запитів.

Усі дії користувача в системі супроводжуються взаємодією з базою даних, що забезпечує збереження актуальної інформації та її подальше використання. Завдяки цьому система підтримує цілісність даних і дозволяє отримувати достовірну інформацію у будь-який момент часу.

Таким чином, логіка роботи розроблюваного веб-додатку базується на чіткій взаємодії між його основними компонентами та забезпечує виконання всіх необхідних операцій із документами. Вона створює основу для ефективної автоматизації документообігу та зручної роботи користувачів із системою.

2.5 Моделювання взаємодії користувача з системою

Моделювання взаємодії користувача з інформаційною системою є важливим етапом проектування, оскільки дозволяє визначити основні сценарії використання програмного продукту та забезпечити зручність його застосування. У розроблюваному веб-додатку для автоматизації документообігу взаємодія користувача із системою організована через веб-інтерфейс та чат-бот, що забезпечує гнучкий підхід до виконання операцій.

Основним суб'єктом взаємодії є користувач, який працює із системою через браузер. Після переходу на головну сторінку користувач може виконати вхід до системи, ввівши свої облікові дані. У разі успішної авторизації він отримує доступ до основного функціоналу, що включає перегляд документів, їх створення та зміну статусу.

Одним із базових сценаріїв є перегляд списку документів. Після входу в систему користувач переходить до панелі керування, де відображається перелік усіх документів із зазначенням їх основних характеристик. Користувач може аналізувати інформацію, відслідковувати статус документів та приймати рішення щодо подальших дій.

Іншим важливим сценарієм є створення нового документа. У цьому випадку користувач відкриває відповідну форму, вводить необхідні дані та надсилає їх на сервер. Система обробляє отриману інформацію, зберігає її у базі даних і відображає новий документ у загальному списку. Такий процес дозволяє швидко додавати нові записи та підтримувати актуальність інформації.

Зміна статусу документа є ще одним типовим сценарієм взаємодії. Користувач обирає потрібний документ та встановлює новий статус залежно від етапу його обробки. Система миттєво оновлює дані, що дозволяє іншим користувачам отримувати актуальну інформацію про стан документообігу.

Окремим сценарієм є взаємодія з чат-ботом, який інтегрований у веб-інтерфейс системи. Користувач може вводити текстові запити, наприклад, отримати інформацію про кількість документів або їх статус. Чат-бот обробляє запит і повертає відповідь, що дозволяє виконувати частину операцій без

використання основного інтерфейсу. Такий підхід спрощує взаємодію із системою та підвищує швидкість доступу до інформації.

Усі сценарії взаємодії користувача із системою побудовані таким чином, щоб забезпечити простоту використання та мінімізувати кількість необхідних дій. Це досягається за рахунок логічної структури інтерфейсу та чіткого розподілу функціоналу між основними компонентами системи.

Таким чином, моделювання взаємодії користувача з системою дозволило визначити ключові сценарії використання веб-додатку та забезпечити їх ефективну реалізацію. Це створює основу для подальшого етапу розробки, який передбачає реалізацію функціоналу системи та перевірку її роботи.

2.6 Проєктування системи ролей та доступів

У процесі проєктування веб-додатку для автоматизації документообігу важливим етапом є розробка системи ролей та доступів. Вона визначає, які дії можуть виконувати користувачі, а також забезпечує захист даних і контроль над операціями в системі. Реалізація такого механізму дозволяє адаптувати функціонал додатку відповідно до ролі кожного користувача.

При проєктуванні системи було обрано рольову модель доступу, яка передбачає розподіл користувачів на кілька категорій залежно від їхніх повноважень. У межах розроблюваного додатку передбачено чотири основні ролі: адміністратор, менеджер, користувач та гість. Такий підхід дозволяє гнучко керувати доступом до функціоналу системи та забезпечити необхідний рівень безпеки.

Роль адміністратора передбачає повний доступ до системи. Користувач із цією роллю має можливість керувати обліковими записами інших користувачів, змінювати їх ролі, створювати, редагувати та видаляти документи, а також контролювати всі процеси, що відбуваються у системі. Адміністратор відповідає за загальне функціонування системи та її налаштування.

Менеджер виконує функції, пов'язані з обробкою документів. Він має право переглядати документи, редагувати їх, змінювати статуси, зокрема погоджувати або відхиляти документи, а також переводити їх у архів. При цьому менеджер не має доступу до керування користувачами, що дозволяє обмежити його повноваження лише роботою з документами.

Звичайний користувач має базовий рівень доступу до системи. Він може створювати нові документи, переглядати існуючі записи та завантажувати прикріплені файли. Водночас користувач не має можливості змінювати статуси документів або редагувати їх після створення, що зменшує ризик внесення некоректних змін.

Роль гостя передбачає мінімальний рівень доступу. Користувач із цією роллю не має можливості працювати з документами і може лише переглядати інформаційне повідомлення про необхідність отримання доступу. Такий підхід дозволяє контролювати нових користувачів та запобігати несанкціонованому доступу до системи.

У межах проєктування було визначено механізм призначення ролей. Після реєстрації кожен користувач автоматично отримує роль гостя. Надання доступу до системи здійснюється адміністратором шляхом зміни ролі користувача(рис. 2.6). Це дозволяє забезпечити додатковий контроль і перевірку перед наданням доступу до даних.

Реалізація системи ролей тісно пов'язана з перевіркою прав доступу під час виконання операцій. Для кожної дії у системі визначається, які ролі мають право її виконувати. Наприклад, редагування документів доступне лише менеджеру та адміністратору, а видалення документів — виключно адміністратору. Такий підхід дозволяє уникнути помилок та несанкціонованих змін.

Крім того, система ролей інтегрована з інтерфейсом користувача. В залежності від ролі змінюється доступний функціонал, зокрема відображення кнопок, форм та можливостей взаємодії з документами. Це підвищує зручність використання системи та запобігає виконанню недоступних операцій.

Керування користувачами

ID	Логін	Поточна роль	Змінити роль
1	антон	admin	admin admin manager user guest Зберегти
2	Евген	manager	manager user guest Зберегти
3	Аня	user	user Зберегти
4	Оля	manager	manager Зберегти
5	Вадим	user	user Зберегти
6	Віталій	guest	guest Зберегти

Рис. 2.6 – Керування користувачами

Таким чином, проєктування системи ролей та доступів дозволило забезпечити чітке розмежування прав користувачів, підвищити рівень безпеки та організувати ефективну роботу з документами. Реалізована модель доступу відповідає поставленим вимогам та забезпечує гнучкість у подальшому розвитку системи.

2.7 Проєктування механізму обробки документів

Одним із ключових етапів проєктування веб-додатку для автоматизації документообігу є розробка механізму обробки документів. Саме цей механізм визначає життєвий цикл документа в системі, порядок його створення, обробки, погодження та завершення. Правильно організований процес обробки дозволяє забезпечити ефективну роботу користувачів та контроль над станом документів.

У процесі проєктування було визначено, що кожен документ проходить кілька послідовних етапів. Початковим етапом є створення документа користувачем. На цьому етапі формується основна інформація, зокрема назва, тип, опис та, за потреби, додається файл. Після створення документ автоматично отримує початковий статус, що дозволяє ідентифікувати його як новий.

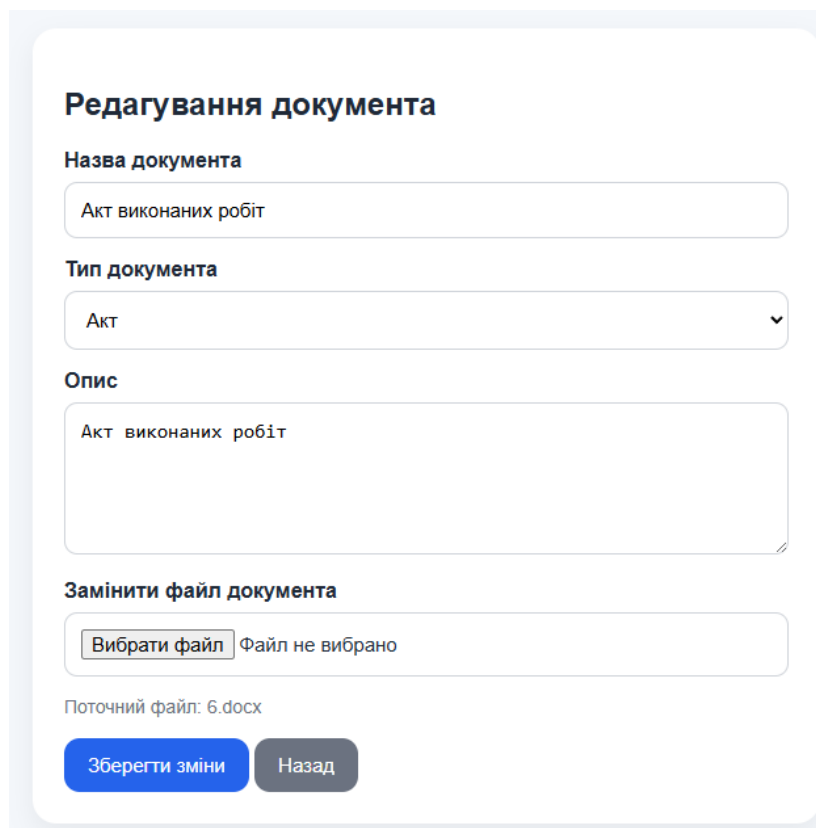
Наступним етапом є обробка документа. На цьому етапі менеджер або адміністратор здійснює перевірку змісту документа, його відповідності встановленим вимогам та приймає рішення щодо подальшої роботи з ним. У результаті обробки документ може бути переведений у стан погодження або відхилення.

Погодження документа є важливим етапом, який свідчить про завершення його перевірки та готовність до використання. У системі передбачено можливість зміни статусу документа на погоджений, що дозволяє відокремити завершені документи від тих, які ще перебувають у роботі.

У випадку, якщо документ не відповідає вимогам або містить помилки, він може бути відхилений. Такий статус дозволяє користувачу зрозуміти, що документ потребує доопрацювання або не може бути використаний у поточному вигляді. Це сприяє підвищенню якості документів та зменшенню кількості помилок.

Окремим етапом є архівація документів. Після завершення роботи документ може бути переведений у статус архіву, що дозволяє зберігати його в системі для подальшого використання або перегляду. Архівування забезпечує впорядкування даних та спрощує роботу з великою кількістю документів.

Важливим аспектом проєктування є можливість редагування документів(рис. 2.7.1). У системі передбачено, що редагування доступне лише для користувачів із відповідними правами. Це дозволяє вносити зміни до документа на етапі його обробки, водночас зберігаючи контроль над цим процесом.



Редагування документа

Назва документа

Акт виконаних робіт

Тип документа

Акт

Опис

Акт виконаних робіт

Замінити файл документа

Вибрати файл Файл не вибрано

Поточний файл: 6.docx

Зберегти зміни Назад

Рис. 2.7.1 – Редагування документа

Крім того, механізм обробки документів включає роботу з файлами. Користувач має можливість прикріплювати файли до документів, а також завантажувати їх для перегляду. Це розширює функціональні можливості системи та дозволяє працювати не лише з текстовими даними, а й із повноцінними документами.

Для забезпечення зручності роботи було передбачено можливість пошуку, фільтрації та сортування документів(рис. 2.7.2). Це дозволяє користувачам швидко знаходити необхідну інформацію, відслідковувати статуси та працювати з документами у межах заданих параметрів, зокрема за датою створення.

Пошук і фільтрація документів

Пошук за назвою:

Тип документа:

Автор:

Статус:

Дата від:

Дата до:

Сортування:

Рис. 2.7.2 – Пошук і фільтрація документів

Таким чином, проєктований механізм обробки документів охоплює всі основні етапи їх життєвого циклу — від створення до архівування. Реалізація такого підходу забезпечує впорядковану роботу з документами, підвищує ефективність взаємодії користувачів та дозволяє контролювати стан кожного документа у системі.

2.8 Проєктування інтерфейсу користувача (UI/UX)

Проєктування інтерфейсу користувача є важливим етапом розробки веб-додатку, оскільки саме через інтерфейс відбувається взаємодія користувача із системою. Від зручності та зрозумілості інтерфейсу залежить ефективність роботи користувачів, швидкість виконання операцій та загальне сприйняття системи [15].

У процесі проєктування інтерфейсу було враховано принципи простоти та логічності. Основні елементи системи організовані таким чином, щоб користувач міг швидко орієнтуватися у функціоналі додатку. Головна сторінка містить навігаційне меню (рис. 2.8.1), яке забезпечує доступ до основних розділів, зокрема панелі керування, створення документів та керування користувачами.

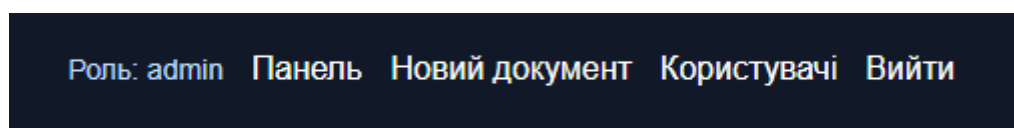


Рис. 2.8.1 – Навігаційне меню

Інтерфейс панелі керування(рис. 2.8.2) є центральним елементом системи. На цій сторінці відображається список документів у вигляді таблиці, що містить основну інформацію про кожен документ, включаючи назву, тип, автора, статус та дату створення. Така форма подання дозволяє швидко переглядати велику кількість записів і оцінювати їхній стан.

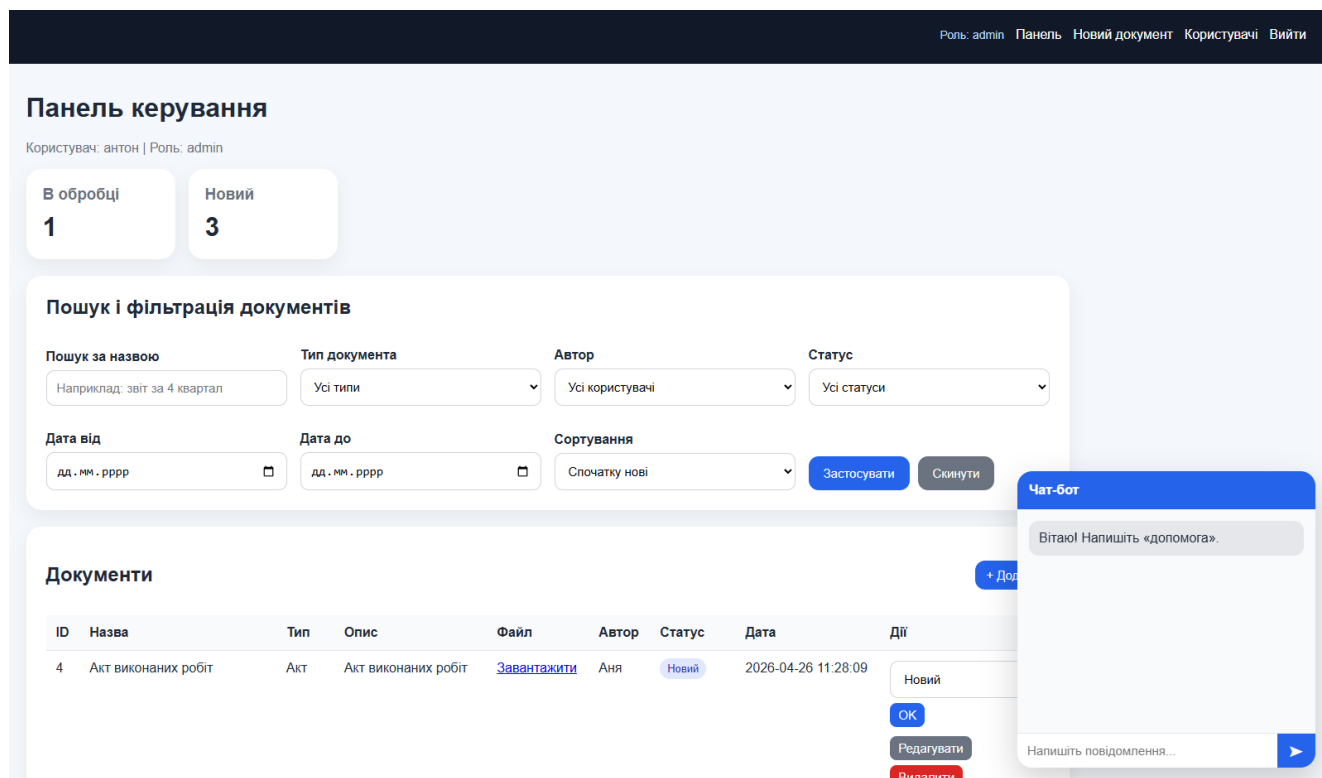


Рис. 2.8.2 – Панель керування

Для підвищення зручності роботи користувачів у системі реалізовано механізми пошуку, фільтрації та сортування. Відповідні елементи інтерфейсу розміщені у верхній частині сторінки та дозволяють задавати параметри відбору документів. Це значно спрощує процес знаходження потрібної інформації, особливо при роботі з великою кількістю даних.

Особливу увагу було приділено формам введення даних. Сторінка створення та редагування документа містить поля для введення назви, вибору типу, додавання опису та завантаження файлів. Усі елементи форми мають зрозумілі підписи, що дозволяє користувачу без труднощів заповнювати необхідну інформацію.

Важливою частиною інтерфейсу є відображення доступних дій відповідно до ролі користувача. Система динамічно змінює набір елементів керування залежно від прав доступу. Наприклад, кнопки редагування або видалення документів доступні лише користувачам із відповідними повноваженнями. Це не лише підвищує безпеку, але й спрощує інтерфейс для користувачів із обмеженим доступом.

У межах проєктування також було враховано необхідність зворотного зв'язку з користувачем. Система відображає повідомлення про результати виконаних дій, наприклад створення або оновлення документа(рис. 2.8.3). Це дозволяє користувачу контролювати свої дії та уникати помилок.

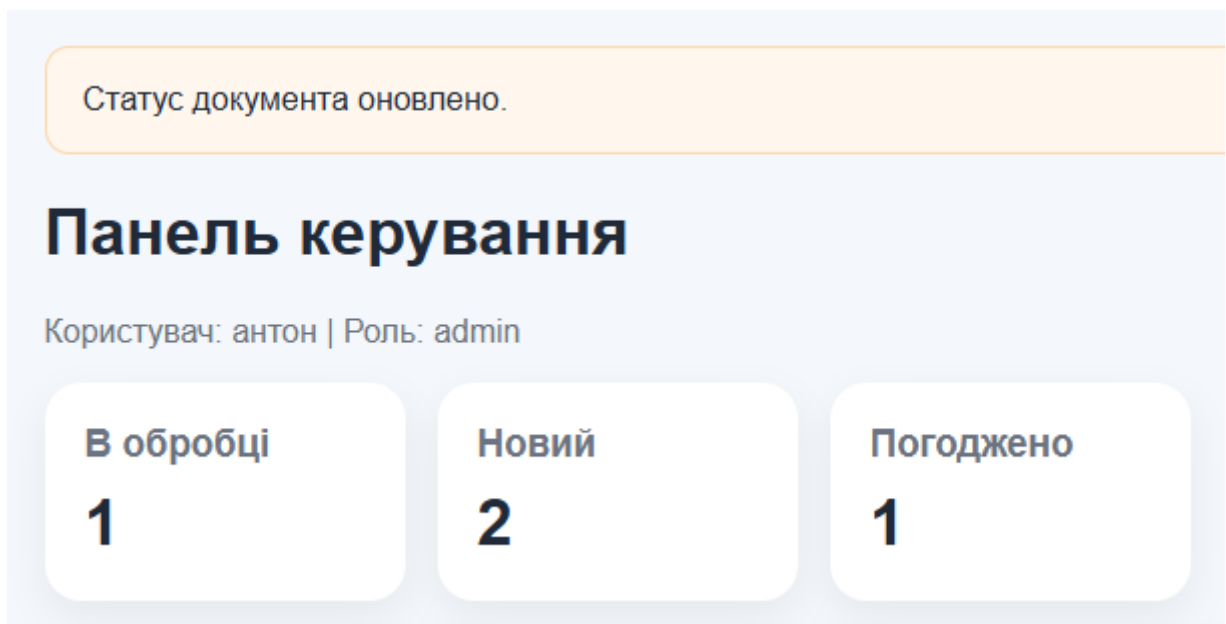


Рис. 2.8.3 – Зворотний зв'язок

Додатковим елементом інтерфейсу є чат-бот, який реалізований у вигляді окремого віджета. Він дозволяє користувачам взаємодіяти із системою за допомогою текстових запитів, отримувати інформацію про документи та виконувати базові дії. Такий підхід робить систему більш сучасною та зручною у використанні.

Таким чином, проєктування інтерфейсу користувача було спрямоване на створення зручного, зрозумілого та функціонального середовища для роботи з

документами. Реалізовані рішення забезпечують ефективну взаємодію користувачів із системою та сприяють підвищенню продуктивності роботи.

2.9 Проєктування механізму пошуку та фільтрації даних

У сучасних інформаційних системах, зокрема у веб-додатках для автоматизації документообігу, важливим елементом є механізм пошуку та фільтрації даних. Зі збільшенням кількості документів у системі виникає необхідність швидкого доступу до потрібної інформації. Саме тому ефективна організація пошуку та відбору даних є важливою складовою проєктування системи.

У межах розроблюваного веб-додатку було передбачено реалізацію комбінованого механізму пошуку та фільтрації, який дозволяє користувачам знаходити документи за різними параметрами. Основним елементом є текстовий пошук за назвою документа. Такий підхід дає змогу швидко знаходити записи за ключовими словами, навіть якщо користувач не пам'ятає точну назву документа [4].

Крім текстового пошуку, система підтримує фільтрацію за кількома параметрами. Користувач може обрати тип документа, його статус або автора. Це дозволяє звужити вибірку та зосередитися лише на тих документах, які відповідають заданим умовам. Такий підхід особливо ефективний у випадках, коли база даних містить значну кількість записів.

Важливим елементом механізму є можливість фільтрації за датами. Користувач може задати період часу, у межах якого були створені документи, що дозволяє аналізувати дані за певний проміжок часу. Це корисно для контролю діяльності організації, формування звітності та відстеження змін у системі.

Окрім фільтрації, у системі реалізовано механізм сортування даних. Користувач має можливість впорядковувати документи за датою створення, назвою або статусом. Це дозволяє організувати відображення інформації у найбільш зручному вигляді залежно від поточних потреб.

Проектування механізму пошуку та фільтрації передбачає також врахування ефективності обробки запитів. Для цього використовуються оптимізовані запити до бази даних, які дозволяють швидко отримувати результати навіть при значному обсязі інформації. Це забезпечує швидкодію системи та комфортну роботу користувачів.

Окремо слід зазначити інтеграцію механізму пошуку з чат-ботом. Користувач може сформулювати запит у текстовому вигляді, після чого система виконує пошук документів за вказаними параметрами. Такий підхід спрощує взаємодію з системою та робить її більш гнучкою у використанні.

Інтерфейс пошуку та фільтрації реалізований у вигляді зручної форми, що містить відповідні поля для введення параметрів. Усі елементи розміщені таким чином, щоб користувач міг швидко налаштувати необхідні умови відбору та отримати результат.

Таким чином, проєктований механізм пошуку та фільтрації даних забезпечує швидкий доступ до інформації, підвищує ефективність роботи користувачів та дозволяє зручно працювати з великою кількістю документів. Реалізація такого механізму є важливою складовою сучасних систем документообігу та відповідає поставленим вимогам до функціональності веб-додатку.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБ-ДОДАТКУ

3.1 Вибір технологій та засобів розробки

Вибір технологій та засобів розробки є важливим етапом створення програмного забезпечення, оскільки від нього залежить ефективність реалізації функціоналу, зручність подальшої підтримки системи та можливість її масштабування. Для розробки веб-додатку автоматизації документообігу з використанням чат-боту було обрано сучасні, але водночас достатньо прості у використанні технології, що дозволяють швидко реалізувати необхідний функціонал.

Основною мовою програмування для реалізації серверної частини системи обрано Python. Це пояснюється його популярністю, зрозумілим синтаксисом та широкими можливостями для створення веб-додатків. Python дозволяє ефективно реалізувати логіку обробки даних, взаємодію з базою даних та обробку запитів користувачів.

Для створення серверної частини веб-додатку використано фреймворк Flask. Даний інструмент є легковаговим і забезпечує необхідний мінімум функціональності для створення веб-додатків. Його використання дозволяє швидко реалізувати маршрутизацію запитів, обробку форм, роботу з сесіями користувачів та інтеграцію з іншими компонентами системи [16].

Клієнтська частина системи реалізована з використанням стандартних веб-технологій, таких як HTML, CSS та JavaScript. HTML використовується для структурування веб-сторінок, CSS — для оформлення інтерфейсу, а JavaScript забезпечує динамічну взаємодію користувача із системою. Зокрема, JavaScript використовується для реалізації роботи чат-боту, що дозволяє надсилати запити на сервер без перезавантаження сторінки [15].

Для збереження даних у системі використовується реляційна база даних SQLite. Це легковагове рішення, яке не потребує окремого серверу баз даних та добре підходить для невеликих і середніх проєктів. Використання SQLite дозволяє

забезпечити надійне збереження даних та простоту інтеграції з серверною частиною додатку [17].

Важливим аспектом є використання механізмів безпеки, зокрема хешування паролів користувачів. Для цього застосовуються відповідні бібліотеки, які дозволяють зберігати паролі у зашифрованому вигляді та забезпечують захист облікових записів від несанкціонованого доступу.

Для розробки та тестування програмного забезпечення використовується середовище розробки, яке забезпечує зручність написання коду, його перевірки та налагодження. Запуск веб-додатку здійснюється на локальному сервері, що дозволяє перевірити його роботу перед подальшим використанням.

Обраний набір технологій забезпечує оптимальне поєднання простоти, функціональності та ефективності. Він дозволяє реалізувати всі необхідні можливості системи, включаючи роботу з документами, авторизацію користувачів та інтеграцію чат-боту, а також створює основу для подальшого розвитку та вдосконалення програмного продукту.

Таким чином, вибір технологій для розробки веб-додатку є обґрунтованим та відповідає поставленим задачам, забезпечуючи ефективну реалізацію функціоналу системи автоматизації документообігу.

3.2 Реалізація функціоналу веб-додатку

Реалізація функціоналу веб-додатку автоматизації документообігу здійснюється на основі розробленої архітектури та визначених вимог до системи. Основна логіка додатку реалізована у серверній частині з використанням мови програмування Python та фреймворку Flask, що забезпечує обробку запитів користувачів і взаємодію з базою даних.

Одним із ключових елементів реалізації є організація маршрутизації запитів. У межах додатку визначено окремі маршрути для кожної функціональної можливості, зокрема для авторизації користувачів, перегляду списку документів, створення нових записів та зміни їх статусу. Кожен маршрут обробляє відповідний

HTTP-запит, виконує необхідні дії та повертає результат у вигляді веб-сторінки або даних [18].

Важливим компонентом системи є реалізація механізму авторизації користувачів, який забезпечує захист даних та контроль доступу до функціоналу веб-додатку. На рисунку 3.2.1 зображена реалізація форми реєстрації користувача, за допомогою якої новий користувач може створити власний обліковий запис для подальшої роботи із системою. Під час реєстрації користувач вводить основні дані, які після перевірки зберігаються у базі даних. Після створення облікового запису користувачу автоматично призначається відповідна роль із визначеним рівнем доступу.

```
@app.route(rule="/register", methods=["GET", "POST"]) 4+ usages
def register():
    if request.method == "POST":
        username = request.form.get("username", "").strip()
        password = request.form.get("password", "").strip()

        if not username or not password:
            flash("Заповніть усі поля.")
            return redirect(url_for("register"))

        conn = get_db()

        try:
            conn.execute(
                sql="INSERT INTO users (username, password_hash, role) VALUES (?, ?, ?)",
                parameters=(username, generate_password_hash(password), "guest")
            )
            conn.commit()
        except sqlite3.IntegrityError:
            flash("Користувач із таким логіном уже існує.")
            return redirect(url_for("register"))
        finally:
            conn.close()

        flash("Акаунт створено. Після входу передайте ваш username керівництву для отримання доступу.")
        return redirect(url_for("login"))

    return render_template("register.html")
```

Рис. 3.2.1 Програмна реалізація реєстрації користувача

На рисунку 3.2.2 зображена реалізація авторизації користувача. Під час входу до системи введені користувачем дані передаються на сервер, де здійснюється перевірка їх відповідності записам у базі даних. У разі успішної перевірки створюється сесія користувача, що дозволяє зберігати інформацію про

його стан під час роботи з системою. Це забезпечує контроль доступу та обмежує виконання дій лише авторизованими користувачами.

```
@app.route(rule: "/login", methods=["GET", "POST"]) 7+ usages
def login():
    if request.method == "POST":
        username = request.form.get("username", "").strip()
        password = request.form.get("password", "").strip()

        conn = get_db()
        user = conn.execute(sql: "SELECT * FROM users WHERE username = ?", parameters: (username,)).fetchone()
        conn.close()

        if user and check_password_hash(user["password_hash"], password):
            session["user_id"] = user["id"]
            session["username"] = user["username"]
            session["role"] = user["role"]

            if user["role"] == "guest":
                return redirect(url_for("guest_wait"))

            return redirect(url_for("dashboard"))

        flash("Невірний логін або пароль.")
        return redirect(url_for("login"))

    return render_template("login.html")
```

Рис. 3.2.2 Програмна реалізація авторизації користувача

Додатково механізм авторизації використовується для реалізації системи ролей користувачів. Залежно від ролі користувача система надає доступ лише до певних функцій веб-додатку. Наприклад, адміністратор має повний доступ до керування системою, менеджер може редагувати та погоджувати документи, а звичайний користувач має доступ лише до створення та перегляду документів. Такий підхід дозволяє підвищити рівень безпеки системи та забезпечити коректний розподіл прав доступу між користувачами.

Функціонал роботи з документами реалізовано через відповідні серверні обробники, які взаємодіють із базою даних. Під час створення нового документа(рис. 3.2.3) система отримує дані з веб-форми, перевіряє їх коректність і зберігає у таблиці документів. Кожен запис містить інформацію про назву документа, його тип, опис, статус та дату створення. Це дозволяє формувати повний перелік документів і відображати його користувачу у зручному вигляді.

```

@app.route(rule: "/documents/new", methods=["GET", "POST"]) 5+ usages
def new_document():
    access_response = require_documents_access()
    if access_response:
        return access_response

    if not can_create_documents():
        flash("У вас немає прав для створення документів.")
        return redirect(url_for("dashboard"))

    if request.method == "POST":
        title = request.form.get("title", "").strip()
        doc_type = request.form.get("doc_type", "").strip()
        description = request.form.get("description", "").strip()
        uploaded_file = request.files.get("document_file")

        if not title or not doc_type:
            flash("Назва та тип документа є обов'язковими.")
            return redirect(url_for("new_document"))

        if doc_type not in DOCUMENT_TYPES:
            flash("Некоректний тип документа.")
            return redirect(url_for("new_document"))

        stored_filename = None
        original_filename = None

        if uploaded_file and uploaded_file.filename:
            if not allowed_file(uploaded_file.filename):
                flash("Непідтримуваний формат файлу. Дозволено: pdf, doc, docx, xls,xlsx, txt, png, jpg.")
                return redirect(url_for("new_document"))

            original_filename = secure_filename(uploaded_file.filename)
            timestamp = datetime.now().strftime("%Y%m%d%H%M%S")
            stored_filename = f"{timestamp}_{session['user_id']}_{original_filename}"
            uploaded_file.save(UPLOAD_FOLDER / stored_filename)

            now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

            conn = get_db()
            conn.execute(
                sql: """
                INSERT INTO documents
                (title, doc_type, description, created_by, file_name, original_file_name, created_at, update
                VALUES (?, ?, ?, ?, ?, ?, ?, ?)
                """, parameters: (
                title, doc_type, description, session["user_id"],
                stored_filename, original_filename, now, now
            ))
            conn.commit()
            conn.close()

            flash("Документ створено.")
            return redirect(url_for("dashboard"))

    return render_template("new_document.html")

```

Рис. 3.2.3 Програмна реалізація створення нового документа

Перегляд документів здійснюється шляхом отримання даних із бази даних та їх передачі у шаблони веб-сторінок. Для цього використовуються спеціальні механізми шаблонізації, які дозволяють динамічно формувати інтерфейс залежно від вмісту бази даних. У результаті користувач отримує актуальну інформацію про всі документи, що зберігаються в системі.

Зміна статусу документа(рис. 3.2.4) реалізована через форму, яка дозволяє обрати новий стан для конкретного запису. Після підтвердження дії відповідна інформація оновлюється у базі даних, що забезпечує актуальність даних та дозволяє відслідковувати процес обробки документа.

```
@app.route(rule: "/documents/<int:doc_id>/status", methods=["POST"]) 1 usage
def update_status(doc_id):
    access_response = require_documents_access()
    if access_response:
        return access_response

    if not can_change_status():
        flash("У вас немає прав для зміни статусу документа.")
        return redirect(url_for("dashboard"))

    status = request.form.get("status", "Новий")

    if status not in STATUSES:
        flash("Некоректний статус документа.")
        return redirect(url_for("dashboard"))

    now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    conn = get_db()
    conn.execute(
        sql: "UPDATE documents SET status = ?, updated_at = ? WHERE id = ?",
        parameters: (status, now, doc_id)
    )
    conn.commit()
    conn.close()

    flash("Статус документа оновлено.")
    return redirect(url_for("dashboard"))
```

Рис. 3.2.4 Програмна реалізація зміни статусу документа

Клієнтська частина додатку реалізована з використанням HTML, CSS та JavaScript. HTML відповідає за структуру сторінок, CSS забезпечує їх оформлення, а JavaScript використовується для реалізації динамічних елементів, зокрема взаємодії з чат-ботом. Завдяки цьому користувач отримує зручний та зрозумілий інтерфейс для роботи із системою.

Таким чином, реалізація функціоналу веб-додатку охоплює всі основні процеси документообігу, включаючи авторизацію користувачів, роботу з документами та відображення інформації. Це дозволяє забезпечити ефективну автоматизацію обробки документів та створює основу для подальшого розвитку системи. На рисунку 3.2.5 наведено структуру розробленого веб-додатку, яка включає серверну частину, шаблони сторінок та статичні ресурси.

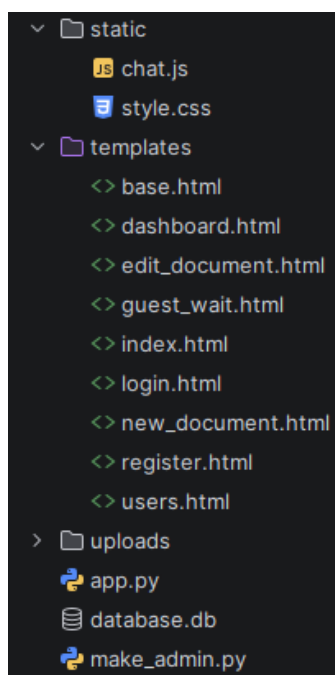


Рис. 3.2.5 – Структура веб-додатку

3.3 Розробка та інтеграція чат-боту

Однією з ключових особливостей розроблюваного веб-додатку є інтеграція чат-боту, який забезпечує альтернативний спосіб взаємодії користувача із системою. Використання чат-боту дозволяє спростити доступ до основного

функціоналу та автоматизувати виконання типових операцій, що підвищує ефективність роботи користувачів.

Розробка чат-боту здійснювалася як частина серверної логіки веб-додатку. Його функціональність реалізована у вигляді окремого обробника запитів, який приймає текстові повідомлення від користувача, аналізує їх та формує відповідь відповідно до заданих правил. Такий підхід дозволяє інтегрувати чат-бот без використання сторонніх сервісів та забезпечити його повну взаємодію з іншими компонентами системи [19].

Взаємодія користувача з чат-ботом реалізована через веб-інтерфейс за допомогою технології асинхронних запитів. Коли користувач вводить повідомлення, воно передається на сервер без перезавантаження сторінки. Сервер обробляє запит, формує відповідь і повертає її у вигляді текстового повідомлення, яке відображається у вікні чату. Такий підхід забезпечує швидку реакцію системи та створює ефект діалогу в режимі реального часу.

Логіка роботи чат-боту(рис. 3.3.1) побудована на основі аналізу ключових слів у введеному повідомленні. Після отримання запиту система перевіряє його зміст на наявність певних фраз або команд, які відповідають за виконання конкретних дій. Наприклад, користувач може отримати інформацію про кількість документів, їх статус або скористатися підказками щодо роботи з системою. У разі відсутності відповідної команди чат-бот повідомляє про необхідність уточнення запиту.

```

elif "документи" in message and "покажи" not in message and "знайди" not in message:
    answer = f"У системі зареєстровано {total_docs} документів."
elif "файли" in message or "файл" in message:
    answer = f"Документів із прикріпленими файлами: {with_files}."
elif "роль" in message:
    answer = f"Ваша роль у системі: {session.get('role', 'guest')}."
elif "типи" in message or "тип" in message:
    answer = "Доступні типи документів:\n" + "\n".join(f"• {item}" for item in DOCUMENT_TYPES)
elif "фільтр" in message or "фільтрація" in message:
    answer = "На панелі керування можна шукати документи за назвою, фільтрувати за типом, автором, статусом і датами, а також сортувати за датою."
elif "статуси" in message or "статус" in message:
    answer = (
        f"Статистика документів:\n"
        f"Нові: {new_docs}\n"
        f"В обробці: {in_work}\n"
        f"Погоджено: {approved}\n"
        f"Відхилено: {rejected}\n"
        f"Архів: {archived}"
    )
elif "створити" in message:
    answer = "Щоб створити документ, натисніть «Новий документ», заповніть назву, тип, опис і за потреби прикріпіть файл."
elif "покажи" in message or "знайди" in message or "побачити" in message:
    query_text = normalize_chat_query(message)
    if not query_text:
        answer = "Напишіть назву документа, наприклад: «покажи документ звіт за 4 квартал»."
    else:

```

Рис. 3.3.1 – Програмна реалізація логіки чат бота

Важливою особливістю чат-боту є його інтеграція з базою даних (рис. 3.3.2). Це дозволяє формувати відповіді на основі актуальної інформації, що зберігається у системі. Наприклад, під час запиту про кількість документів чат-бот виконує відповідний запит до бази даних та повертає користувачу актуальні дані. Такий підхід забезпечує коректність інформації та підвищує практичну цінність чат-боту.

```

conn = get_db()
total_docs = conn.execute("SELECT COUNT(*) as count FROM documents").fetchone()["count"]
new_docs = conn.execute("SELECT COUNT(*) as count FROM documents WHERE status = 'Новий').fetchone()["count"]
in_work = conn.execute("SELECT COUNT(*) as count FROM documents WHERE status = 'В обробці').fetchone()["count"]
approved = conn.execute("SELECT COUNT(*) as count FROM documents WHERE status = 'Погоджено').fetchone()["count"]
rejected = conn.execute("SELECT COUNT(*) as count FROM documents WHERE status = 'Відхилено').fetchone()["count"]
archived = conn.execute("SELECT COUNT(*) as count FROM documents WHERE status = 'Архів').fetchone()["count"]
with_files = conn.execute("SELECT COUNT(*) as count FROM documents WHERE file_name IS NOT NULL").fetchone()["count"]

```

Рис. 3.3.2 – Програмна реалізація інтеграції чат бота з базою даних

Інтеграція чат-боту у веб-додаток здійснена таким чином, що він виступає як допоміжний інструмент, який доповнює основний інтерфейс. Користувач може виконувати ті самі дії як через традиційний інтерфейс, так і за допомогою текстових команд. Це забезпечує гнучкість використання системи та дозволяє обрати найбільш зручний спосіб взаємодії.

Незважаючи на відносну простоту реалізації, розроблений чат-бот виконує всі необхідні функції для автоматизації базових операцій. У подальшому його

функціональність може бути розширена шляхом додавання нових команд або використання більш складних алгоритмів обробки природної мови.

Таким чином, розробка та інтеграція чат-боту дозволили підвищити зручність використання веб-додатку, автоматизувати обробку запитів користувачів та створити більш сучасний інтерфейс взаємодії із системою.

3.4 Реалізація системи авторизації та ролей доступу

Реалізація системи авторизації та ролей доступу є важливим етапом розробки веб-додатку, оскільки саме вона забезпечує контроль доступу користувачів до функціоналу системи та захист даних. У межах розроблюваного додатку було реалізовано механізм автентифікації користувачів, а також рольову модель доступу, яка визначає їхні можливості.

Авторизація користувачів здійснюється за допомогою введення логіну та пароля. Під час реєстрації користувач вводить свої облікові дані, які зберігаються у базі даних. При цьому пароль не зберігається у відкритому вигляді, а обробляється за допомогою алгоритмів хешування, що забезпечує додатковий рівень захисту [2].

Після успішного входу в систему створюється сесія користувача, яка дозволяє ідентифікувати його під час подальшої роботи. У сесії зберігаються основні дані, зокрема ідентифікатор користувача, його ім'я та роль. Це дає можливість системі визначати права доступу до різних функцій без необхідності повторної автентифікації.

У реалізованій системі передбачено чотири ролі користувачів: адміністратор, менеджер, користувач та гість. Після реєстрації кожному користувачу автоматично присвоюється роль гостя. Такий підхід дозволяє обмежити доступ до функціоналу системи до моменту підтвердження прав користувача.

Адміністратор має повний доступ до всіх можливостей системи. Він може керувати користувачами, змінювати їх ролі, створювати, редагувати та видаляти

документи, а також контролювати всі процеси. Менеджер має доступ до функцій обробки документів, включаючи зміну статусів, редагування та переведення документів у архів. Звичайний користувач може створювати документи, переглядати їх та завантажувати файли, але не має прав на їх зміну або видалення.

Користувач із роллю гостя не має доступу до функціоналу роботи з документами. Після входу в систему він бачить повідомлення про необхідність отримання дозволу від адміністратора. Це дозволяє забезпечити контроль над доступом та уникнути несанкціонованого використання системи.

Реалізація ролей доступу здійснюється шляхом перевірки прав користувача перед виконанням кожної операції(рис. 3.4). Для цього у програмному коді передбачено відповідні умови, які визначають, чи має користувач право виконувати певну дію. Наприклад, зміна статусу документа доступна лише менеджеру та адміністратору, а видалення документів — лише адміністратору.

```
def is_guest(): 4+ usages
    return current_role() == "guest"

def is_admin(): 1 usage
    return current_role() == "admin"

def can_access_documents(): 2+ usages
    return current_role() in ("admin", "manager", "user")

def can_create_documents(): 2+ usages
    return current_role() in ("admin", "manager", "user")

def can_change_status(): 2+ usages
    return current_role() in ("admin", "manager")

def can_edit_documents(): 2+ usages
    return current_role() in ("admin", "manager")

def can_delete_documents(): 2+ usages
    return current_role() == "admin"

def can_manage_users(): 2+ usages
    return current_role() == "admin"
```

Рис. 3.4 – Програмна реалізація ролей доступу

Крім серверної перевірки прав доступу, було реалізовано також обмеження на рівні інтерфейсу користувача. Залежно від ролі користувача відображаються або приховуються відповідні елементи керування, такі як кнопки редагування чи видалення. Це дозволяє зробити інтерфейс більш зрозумілим і уникнути виконання недоступних операцій.

Таким чином, реалізована система авторизації та ролей доступу забезпечує надійний контроль над діями користувачів, підвищує рівень безпеки та дозволяє ефективно розмежувати права доступу. Це є важливою складовою функціонування веб-додатку та відповідає сучасним вимогам до інформаційних систем.

3.5 Реалізація роботи з файлами документів

Однією з важливих складових розробленого веб-додатку є реалізація роботи з файлами документів. У системах документообігу значна частина інформації представлена у вигляді файлів різних форматів, тому забезпечення можливості їх завантаження, збереження та перегляду є необхідною умовою ефективної роботи користувачів.

У межах розробленого додатку реалізовано механізм завантаження файлів під час створення або редагування документа. Користувач має можливість обрати файл із локального пристрою та прикріпити його до відповідного запису. Для цього використовується стандартний механізм передачі файлів через веб-форму, що дозволяє зручно взаємодіяти із системою без використання додаткових інструментів [18].

Під час завантаження файлів здійснюється перевірка їх формату. У системі дозволено використання лише певних типів файлів, зокрема текстових документів, електронних таблиць, зображень та PDF-файлів. Такий підхід дозволяє запобігти завантаженню небезпечного або невідомого контенту, що підвищує рівень безпеки системи.

Завантажені файли зберігаються на сервері у спеціально виділеному каталозі. Для уникнення конфліктів імен файлів використовується унікальне найменування, яке формується на основі часу завантаження та ідентифікатора користувача. Це забезпечує коректне збереження файлів навіть у випадку, коли різні користувачі завантажують файли з однаковими назвами.

Інформація про прикріплений файл, зокрема його ім'я та шлях до збереження, записується у базу даних разом із даними про документ. Це дозволяє пов'язати файл із відповідним записом та забезпечити швидкий доступ до нього у майбутньому.

Для користувачів реалізовано можливість завантаження прикріплених файлів. У інтерфейсі системи відображається відповідне посилання, яке дозволяє скачати файл на локальний пристрій. Доступ до цієї функції обмежений відповідно до ролі користувача, що забезпечує контроль над розповсюдженням інформації.

Окрему увагу приділено можливості оновлення файлів під час редагування документа. Користувач із відповідними правами може замінити існуючий файл новим, що дозволяє підтримувати актуальність інформації. При цьому старий файл може бути замінений або збережений залежно від реалізації системи.

Реалізація роботи з файлами також враховує питання безпеки. Перед завантаженням файлів перевіряється їх тип, а доступ до файлів обмежується лише авторизованими користувачами. Це дозволяє запобігти несанкціонованому доступу та використанню файлів сторонніми особами.

Таким чином, реалізований механізм роботи з файлами забезпечує повний цикл взаємодії з документами, включаючи їх завантаження, збереження та перегляд. Це розширює функціональні можливості системи та робить її більш практичною для використання у реальних умовах.

3.6 Реалізація пошуку та фільтрації документів

У процесі розробки веб-додатку для автоматизації документообігу важливим етапом стала реалізація механізму пошуку та фільтрації документів. Даний функціонал забезпечує швидкий доступ до необхідної інформації та значно підвищує ефективність роботи користувачів із системою.

Реалізація пошуку документів базується на використанні текстового введення, що дозволяє здійснювати пошук за назвою документа. Користувач може ввести ключове слово або частину назви, після чого система виконує відповідний запит до бази даних та повертає результати, які відповідають заданим критеріям. Такий підхід забезпечує гнучкість і зручність використання, оскільки не вимагає точного введення повної назви документа.

Фільтрація документів реалізована за кількома параметрами, що дозволяє користувачеві звужити вибірку даних. Зокрема, передбачено можливість відбору документів за типом, статусом та автором. Користувач може обрати один або декілька параметрів одночасно, що дає змогу сформувати більш точний результат пошуку.

Окремо реалізовано фільтрацію за датами створення документів. Користувач має можливість задати початкову та кінцеву дату, що дозволяє переглядати документи лише за визначений період. Такий функціонал є корисним для аналізу діяльності за певний часовий проміжок та формування звітності.

Результати пошуку та фільтрації відображаються у вигляді таблиці, що містить основну інформацію про документи. Усі зміни параметрів пошуку виконуються без необхідності переходу на інші сторінки, що забезпечує зручність та швидкість взаємодії з системою.

Для покращення роботи з даними реалізовано механізм сортування результатів. Користувач може обрати порядок відображення документів, зокрема за датою створення, назвою або статусом. Це дозволяє організувати інформацію відповідно до поточних потреб користувача.

Особливістю реалізації є використання комбінованих запитів до бази даних, які враховують усі задані параметри. Це дозволяє отримувати точні результати

навіть при одночасному використанні кількох фільтрів. Завдяки цьому система забезпечує високу ефективність обробки запитів.

Додатково реалізовано інтеграцію пошуку з чат-ботом. Користувач може сформулювати запит у природній формі, після чого система аналізує його та виконує пошук документів за відповідними параметрами. Наприклад, при введенні запиту з назвою документа чат-бот повертає список знайдених результатів. Це спрощує взаємодію з системою та робить її більш інтуїтивно зрозумілою.

Таким чином, реалізований механізм пошуку та фільтрації документів забезпечує швидкий і зручний доступ до інформації, дозволяє ефективно працювати з великою кількістю даних та підвищує загальну продуктивність системи. Це є важливим компонентом сучасного веб-додатку для автоматизації документообігу.

3.7 Аналіз продуктивності системи

Аналіз продуктивності системи є важливим етапом оцінки розробленого веб-додатку, оскільки дозволяє визначити ефективність його роботи та здатність обробляти запити користувачів у реальних умовах експлуатації. У межах розробки було проведено оцінку швидкодії основних функціональних компонентів системи, зокрема роботи з базою даних, обробки запитів та взаємодії з користувачем.

Одним із ключових факторів продуктивності є швидкість обробки запитів до бази даних. У системі використовується реляційна база даних, яка забезпечує збереження інформації про користувачів та документи. Для підвищення ефективності роботи застосовано оптимізовані SQL-запити, що дозволяють виконувати пошук, фільтрацію та сортування даних без значних затримок. Навіть при використанні кількох параметрів фільтрації система демонструє стабільну швидкість відповіді.

Особливу увагу було приділено продуктивності механізму пошуку документів. Завдяки використанню часткового співпадіння рядків користувач

може швидко знаходити необхідні документи, а система обробляє такі запити без суттєвого навантаження. Це забезпечує зручність роботи навіть при збільшенні кількості записів у базі даних.

Важливим аспектом є також обробка файлів. Завантаження та скачування документів здійснюється через серверну частину додатку. Для запобігання перевантаженню системи обмежено допустимі типи файлів, що дозволяє зменшити ризик обробки великих або небезпечних даних. Збереження файлів у файловій системі, а не безпосередньо у базі даних, позитивно впливає на продуктивність та спрощує доступ до них.

Оцінка продуктивності інтерфейсу користувача показала, що система швидко реагує на дії користувача. Завдяки використанню сучасних підходів до побудови веб-інтерфейсу забезпечується плавна робота елементів управління та швидке оновлення інформації. Це створює комфортні умови для взаємодії з додатком.

Окремо було проаналізовано роботу чат-боту. Обробка текстових запитів користувачів здійснюється на сервері та не потребує значних обчислювальних ресурсів. Чат-бот використовує просту логіку аналізу повідомлень, що дозволяє забезпечити швидку відповідь без затримок.

Загалом система демонструє стабільну роботу при одночасному виконанні кількох операцій. Навіть при збільшенні кількості користувачів та документів продуктивність залишається на прийнятному рівні. Це свідчить про правильний вибір архітектури та технологій розробки.

Таким чином, проведений аналіз продуктивності показав, що розроблений веб-додаток забезпечує швидку обробку даних, стабільну роботу та зручну взаємодію з користувачем. Отримані результати підтверджують ефективність реалізованих рішень та можливість подальшого масштабування системи.

3.8 Перспективи розвитку системи

Розроблений веб-додаток для автоматизації документообігу є базовим рішенням, яке реалізує основні функції роботи з документами, проте має значний потенціал для подальшого розвитку. Удосконалення системи дозволить підвищити її функціональність, ефективність та адаптивність до реальних умов використання в організаціях різного масштабу.

Одним із напрямів розвитку є розширення функціоналу обробки документів. Зокрема, можливе впровадження багаторівневої системи погодження, яка передбачає проходження документа через декілька етапів перевірки різними користувачами. Це дозволить більш точно відобразити реальні бізнес-процеси організацій.

Перспективним є також впровадження технологій автоматичного розпізнавання тексту у документах. Використання таких рішень дозволить обробляти скановані документи, витягувати з них інформацію та автоматично заповнювати відповідні поля у системі. Це значно скоротить час обробки документів і зменшить вплив людського фактору.

Ще одним важливим напрямом є інтеграція системи з іншими інформаційними сервісами. Це можуть бути системи електронного підпису, поштові сервіси або корпоративні інформаційні системи. Така інтеграція дозволить розширити можливості додатку та зробити його частиною єдиного інформаційного середовища організації.

Удосконалення чат-боту також є перспективним напрямом розвитку. Зокрема, можливе використання технологій штучного інтелекту для більш глибокого аналізу запитів користувачів та автоматичного виконання складніших операцій. Це дозволить зробити взаємодію з системою більш природною та зручною.

Важливим аспектом є підвищення рівня безпеки системи. У майбутньому може бути реалізовано додаткові механізми захисту, такі як двофакторна автентифікація, шифрування даних та розширене логування дій користувачів. Це дозволить забезпечити ще більший рівень захисту інформації.

Крім того, можливе покращення інтерфейсу користувача з урахуванням досвіду використання системи. Це включає оптимізацію розташування елементів, додавання нових візуальних компонентів та підвищення зручності роботи на мобільних пристроях.

Таким чином, розроблена система має значний потенціал для подальшого розвитку та вдосконалення. Реалізація зазначених напрямів дозволить розширити її функціональні можливості, підвищити ефективність роботи та адаптувати систему до потреб сучасних організацій.

3.9 Тестування системи

Тестування є невід'ємним етапом розробки програмного забезпечення, який дозволяє перевірити правильність роботи системи, виявити можливі помилки та оцінити її відповідність поставленим вимогам. У межах розробки веб-додатку для автоматизації документообігу було проведено функціональне тестування основних компонентів системи.

Процес тестування передбачав перевірку роботи ключових функцій веб-додатку в умовах, наближених до реального використання. Особливу увагу було приділено перевірці коректності обробки даних, стабільності роботи серверної частини та взаємодії між окремими модулями системи.

На першому етапі було протестовано механізм авторизації користувачів. Перевірка включала введення правильних і неправильних облікових даних з метою оцінки реакції системи. У результаті встановлено, що система коректно ідентифікує користувачів, забезпечує доступ до функціоналу лише після успішного входу та запобігає несанкціонованому доступу.

Далі було проведено тестування функціоналу роботи з документами. Перевірялася можливість створення нових документів, їх збереження у базі даних та відображення у списку. Результати тестування показали, що всі введені дані коректно обробляються та зберігаються, а створені документи одразу стають доступними для перегляду.

Окрему увагу було приділено перевірці механізму зміни статусу документів. Під час тестування виконувалося послідовне оновлення статусів, що дозволило оцінити коректність роботи відповідного функціоналу. У результаті встановлено, що система правильно оновлює інформацію та відображає актуальний стан документів.

Також було протестовано роботу чат-боту, який є важливим елементом системи. Перевірка включала введення різних типів запитів, зокрема стандартних команд та довільних повідомлень. У ході тестування встановлено, що чат-бот коректно обробляє запити, надає відповіді відповідно до закладеної логіки та взаємодіє з базою даних для отримання актуальної інформації.

Додатково було оцінено зручність використання системи та швидкість її роботи. Встановлено, що інтерфейс є зрозумілим для користувача, а основні операції виконуються без помітних затримок. Це свідчить про відповідність системи нефункціональним вимогам, зокрема щодо продуктивності та зручності використання.

Таким чином, результати тестування підтверджують працездатність розробленого веб-додатку та його відповідність поставленим задачам. Виявлені недоліки мають незначний характер і можуть бути усунені у процесі подальшого вдосконалення системи.

3.10 Оцінка ефективності розробленого рішення

Оцінка ефективності розробленого веб-додатку є важливим етапом завершення проєкту, оскільки дозволяє визначити рівень досягнення поставленої мети та відповідність отриманого результату початковим вимогам. У межах даної роботи ефективність системи оцінюється з урахуванням її функціональних можливостей, зручності використання та впливу на процеси документообігу.

Розроблений веб-додаток забезпечує автоматизацію основних операцій із документами, включаючи їх створення, перегляд та зміну статусу. Це дозволяє значно скоротити час обробки інформації у порівнянні з традиційними підходами,

де значна частина операцій виконується вручну. Завдяки централізованому зберіганню даних користувачі отримують швидкий доступ до необхідної інформації, що підвищує оперативність прийняття рішень.

Важливою перевагою системи є її реалізація у вигляді веб-додатку, що забезпечує доступ до функціоналу через браузер без необхідності встановлення додаткового програмного забезпечення. Це дозволяє використовувати систему з різних пристроїв та значно спрощує її впровадження у діяльність організації.

Інтеграція чат-боту є суттєвим фактором підвищення ефективності роботи системи. Завдяки можливості виконання окремих операцій за допомогою текстових запитів зменшується кількість дій, які користувач повинен виконувати через інтерфейс. Це особливо важливо для типових задач, таких як отримання інформації про документи або їх статус. У результаті підвищується швидкість взаємодії з системою та зручність її використання.

Оцінка зручності інтерфейсу показала, що система має зрозумілу структуру та не потребує тривалого навчання користувачів. Основні функції розміщені у логічних розділах, що дозволяє швидко орієнтуватися у функціоналі додатку. Це сприяє зниженню кількості помилок під час роботи та підвищенню загальної продуктивності користувачів.

З технічної точки зору система демонструє стабільну роботу та достатній рівень продуктивності. Обробка запитів відбувається без значних затримок, що забезпечує комфортну взаємодію з додатком. Використання сучасних технологій дозволяє забезпечити надійність зберігання даних та можливість подальшого розширення функціоналу.

У порівнянні з існуючими рішеннями розроблений веб-додаток має перевагу у простоті реалізації та використання. Він не потребує складного налаштування, має мінімальні вимоги до ресурсів та може бути адаптований під конкретні потреби користувачів. Це робить його доцільним для використання в невеликих організаціях або як прототип більш складної системи.

Таким чином, проведена оцінка показала, що розроблений веб-додаток ефективно вирішує поставлену задачу автоматизації документообігу. Він

забезпечує підвищення швидкості обробки документів, зручність взаємодії користувачів із системою та можливість подальшого розвитку програмного продукту.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було розглянуто актуальну задачу автоматизації документообігу з використанням сучасних інформаційних технологій. Проведений аналіз предметної області дозволив визначити основні проблеми традиційних підходів до організації роботи з документами, зокрема значні витрати часу на обробку інформації, складність контролю стану документів та залежність від людського фактору [1].

У ході дослідження було проаналізовано сучасні системи автоматизації документообігу та встановлено, що, незважаючи на їх широкі функціональні можливості, вони часто є складними у впровадженні та використанні, особливо для невеликих організацій. Це підтвердило доцільність розробки більш простого та доступного веб-додатку, який забезпечує виконання основних операцій із документами.

На основі проведеного аналізу було сформовано вимоги до системи та розроблено її архітектуру. У межах роботи було спроектовано структуру бази даних, визначено основні компоненти системи та описано логіку їх взаємодії. Особливу увагу приділено інтеграції чат-боту як інструменту спрощення взаємодії користувача із системою.

У практичній частині роботи реалізовано веб-додаток для автоматизації документообігу, який забезпечує можливість створення, перегляду та обробки документів, а також контроль їх стану. Додатково було розроблено чат-бот, що дозволяє виконувати окремі операції та отримувати інформацію за допомогою текстових запитів. Це підвищує зручність використання системи та скорочує час виконання типових дій.

Проведене тестування підтвердило працездатність розробленого програмного продукту та його відповідність поставленим вимогам. Система коректно обробляє дані, забезпечує стабільну роботу та має зручний інтерфейс для користувачів. Отримані результати свідчать про можливість практичного застосування розробленого рішення.

Таким чином, у роботі досягнуто поставленої мети, яка полягала у розробці веб-додатку для автоматизації документообігу з використанням чат-боту. Створена система дозволяє підвищити ефективність обробки документів, забезпечити зручну взаємодію користувачів із програмним продуктом та створює основу для подальшого розвитку і вдосконалення функціоналу.

Список використаних джерел

1. Sommerville I. Software Engineering. 10th ed. – Boston: Pearson, 2020. – 816 p.
2. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. – 9th ed. – New York: McGraw-Hill, 2020. – 880 p.
3. Fowler M. Patterns of Enterprise Application Architecture. – Boston: Addison-Wesley, 2021. – 533 p.
4. Silberschatz A., Korth H., Sudarshan S. Database System Concepts. – 7th ed. – New York: McGraw-Hill, 2020. – 1376 p.
5. Elmasri R., Navathe S. Fundamentals of Database Systems. – 7th ed. – Boston: Pearson, 2021. – 1272 p.
6. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. – Boston: Addison-Wesley, 2020. – 395 p.
7. Козак В. М. Інформаційні системи і технології в управлінні. – Київ: Центр учбової літератури, 2021. – 344 с.
8. Биков В. Ю. Інформаційні технології в освіті та управлінні. – Київ: Атіка, 2020. – 280 с.
9. Литвин В. В. Системи підтримки прийняття рішень. – Львів: Новий Світ-2000, 2021. – 312 с.
10. Кравченко О. В. Основи проєктування інформаційних систем. – Харків: ХНУРЕ, 2022. – 256 с.
11. Мельник А. О. Організація баз даних та знань. – Київ: Видавництво Ліра-К, 2020. – 360 с.
12. ISO/IEC 25010:2011. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE).
13. ISO/IEC 12207:2017. Systems and software engineering – Software life cycle processes.
14. Welling L., Thomson L. PHP and MySQL Web Development. – 5th ed. – Addison-Wesley, 2020. – 768 p.
15. Duckett J. HTML and CSS: Design and Build Websites. – Wiley, 2021. – 490 p.

- 16.Flask Documentation. – [Электронный ресурс]. – Режим доступа:
<https://flask.palletsprojects.com/>
- 17.SQLite Documentation. – [Электронный ресурс]. – Режим доступа:
<https://www.sqlite.org/docs.html>
- 18.Mozilla Developer Network. Web Technologies Documentation. – [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/>
- 19.W3Schools Online Web Tutorials. – [Электронный ресурс]. – Режим доступа:
<https://www.w3schools.com/>
- 20.IBM Cloud. What is Document Management? – [Электронный ресурс]. –
Режим доступа: <https://www.ibm.com/>
- 21.Microsoft Learn. Web Development Fundamentals. – [Электронный ресурс]. –
Режим доступа: <https://learn.microsoft.com/>
- 22.Oracle. Database Concepts. – [Электронный ресурс]. – Режим доступа:
<https://docs.oracle.com/>

ПЕРЕЛІК ДЕМОНСТРАЦІЙНИХ МАТЕРІАЛІВ (Презентація)



Тема, мета та актуальність

Тема:

Веб-додаток для автоматизації документообігу з використанням чат-боту

Мета:

Розробка веб-додатку для автоматизації роботи з документами, який забезпечує керування документами, ролями доступу та взаємодію через чат-бот.

Актуальність:

багато організацій досі працюють вручну складність пошуку документів необхідність автоматизації процесів

Веб-додаток для автоматизації документообігу

Система дозволяє створювати, завантажувати, шукати, фільтрувати, сортувати, редагувати та контролювати документи за допомогою веб-інтерфейсу, рівнів доступу та чат-боту.

Увійти

Створити акаунт

Стартова сторінка веб додатку

Вхід

Логін

Пароль

Увійти

Доступ очікує підтвердження

Ваш акаунт має роль **guest**, тому доступ до документів тимчасово обмежено.

Передайте ваш username керівництву або адміністратору системи для отримання доступу.

Віталій

Реєстрація

Логін

Пароль

Зареєструватися

Після реєстрації акаунт отримує роль guest. Для доступу передайте username керівництву.

Реєстрація та вхід

Документи

+ Док

ID	Назва	Тип	Опис	Файл	Автор	Статус	Дата	Дії
4	Акт виконаних робіт	Акт	Акт виконаних робіт	Завантажити	Аня	Новий	2026-04-26 11:28:09	Перегляд
3	Наказ 1	Наказ	Наказ про відпустку	Завантажити	Аня	Новий	2026-04-26 11:27:41	Перегляд
2	Заява на роботу Поповійчук	Заява	Заява від Поповійчук	Завантажити	Евген	В обробці	2026-04-26 11:26:26	Перегляд
1	Звіт за 4 квартал	Звіт	Звіт по зарплаті	Завантажити	Евген	Новий	2026-04-26 11:25:47	Перегляд

Список документів

Створення документа

Назва документа

Наприклад: Звіт за 4 квартал

Тип документа

Заява

Опис

Короткий опис документа

Файл документа

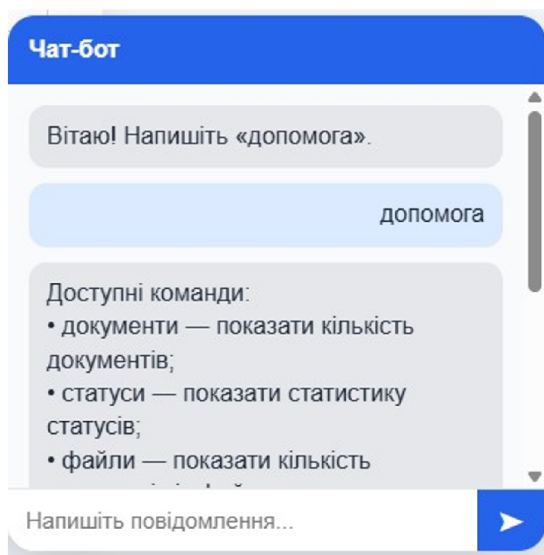
Вибрати файл

Файл не вибрано

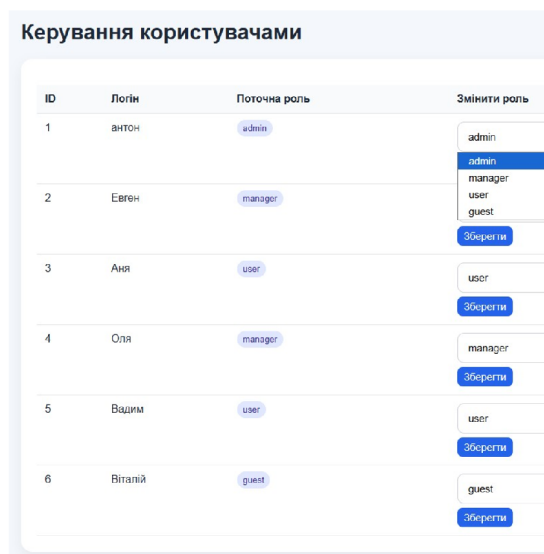
Дозволені формати: PDF, DOC, DOCX, XLS, XLSX, TXT, PNG, JPG.

Створити

Створення документа



Чат бот



Керування користувачами та ролі

Пошук і фільтрація документів

Пошук за назвою

Тип документа

Автор

Статус

Наприклад: звіт за 4 квартал

Усі типи

Усі користувачі

Усі статуси

Дата від

Дата до

Сортування

дд. мм. рrrr

дд. мм. рrrr

Спочатку нові

Застосувати

Скинути

Пошук і фільтрація документів

Панель керування

Користувач: антон | Роль: admin

В обробці

Новий

1

3

Пошук і фільтрація документів

Пошук за назвою

Тип документа

Автор

Статус

Наприклад: звіт за 4 квартал

Усі типи

Усі користувачі

Усі статуси

Дата від

Дата до

Сортування

дд. мм. рrrr

дд. мм. рrrr

Спочатку нові

Застосувати

Скинути

Документи

ID

Назва

Тип

Опис

Файл

Автор

Статус

Дата

Дії

4

Акт виконаних робіт

Акт

Акт виконаних робіт

[Завантажити](#)

Аня

Новий

2026-04-26 11:28:09

Новий

OK

Редагувати

Видалити

Чат-бот

Вітаю! Напишіть «допомога».

Напишіть повідомлення...

Висновки

У результаті роботи:

- створено веб-додаток
- реалізовано систему ролей
- реалізовано роботу з файлами
- інтегровано чат-бот
- реалізовано пошук і фільтрацію

Система може використовуватись в організаціях для автоматизації документообігу.

Апробація результатів та публікації

VII Міжнародна науково-технічна конференція «Сучасний стан та перспективи розвитку IoT» з тезами на тему «Веб-додаток для автоматизації документообігу з використанням чат-боту»

Всеукраїнська науково-технічна конференція «Технології цифрового розвитку: програмні системи та децентралізація» з тезами на тему «Система інтелектуальної маршрутизації цифрових документів на основі чат-інтерфейсів»

Дякую за увагу

