

КВАЛІФІКАЦІЙНА РОБОТА

Система автоматизованої генерації та перевірки типових договорів на основі
введених реквізитів та шаблонів

Назва теми

Рівень вищої освіти перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 126 «Інформаційні системи та технології»

Шифр, назва

Освітня програма «Інформаційні системи та технології»

Назва

Шифр КвРКІ 230156.23.01.05 ПЗ

Виконав здобувач III курсу, група ІСТс-23-1


Підпис

Назар МИХАЙЛОВ
Ініціали, прізвище

Керівник

Науковий ступінь, учене звання


Підпис

Олег ВОЙЧУР
Ініціали, прізвище

Нормоконтролер канд. фіз. мат. наук, доц.

Науковий ступінь, учене звання


Підпис

Тетяна КИСІЛЬ
Ініціали, прізвище

До захисту допускаю:
завідувач кафедри КІС
«01» червня 2026 р.


Підпис

Ольга ПАВЛОВА
Ініціали, прізвище

дата

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)


Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Освітня програма «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Завідувачка кафедри КПС

 Ольга ПАВЛОВА

“ 10 ” 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Михайлову Назару Олександровичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Система автоматизованої генерації та перевірки типових договорів на основі введених реквізитів та шаблонів

Керівник проекту (роботи) Войчур Олег Юрійович, Ст. в.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. № 7

2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Аналіз предметної області та постановка задачі _____

Проектування інформаційної системи автоматизованої генерації та перевірки договорів _____

Реалізація інформаційної системи автоматизованої генерації та перевірки договорів _____

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Архітектура ПЗ проекту _____

Архітектура інформаційної системи _____

Алгоритм роботи системи _____

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання « 10 » 01 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2026	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2026	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2026	виконано
4	Робота над розділом 2 – вибір компонентів для проектування системи автоматизованої генерації та перевірки типових договорів	01.04.2026	виконано
5	Робота над розділом 3 – проектування системи автоматизованої генерації та перевірки типових договорів	29.04.2026	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2026	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач  Підпис Назар МИХАЙЛОВ
Імя, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи  Підпис Олег ВОЙЧУР
Імя, ПРІЗВИЩЕ

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Система автоматизованої генерації та перевірки типових договорів на основі введених реквізитів та шаблонів».

Автор роботи: Назар МИХАЙЛОВ.

Керівник роботи: Олег ВОЙЧУР.

Пояснювальна записка: 64 с., 15 рис., 1 табл., 3 дод., 53 джерел.

Графічна частина: 3 креслення.

АВТОМАТИЗАЦІЯ, ВАЛДАЦІЯ, ДОГОВІР, ДОКУМЕНТ,
ІНФОРМАЦІЙНА СИСТЕМА, РЕКВІЗИТИ, ШАБЛОН.

Кваліфікаційна робота бакалавра присвячена розробці та дослідженню інформаційної системи автоматизованої генерації та перевірки типових договорів на основі введених реквізитів та шаблонів. Актуальність теми зумовлена потребою підвищення швидкості підготовки договірної документації, зменшення кількості помилок під час заповнення реквізитів, уніфікації структури документів і зниження залежності від ручної праці.

Метою роботи є проектування, реалізація та тестування інформаційної системи, призначеної для введення реквізитів, перевірки їх правильності, автоматичного заповнення шаблонів і формування готових договорів у зручному форматі. Для досягнення поставленої мети виконано аналіз сучасних підходів до формування та обробки договорів, обґрунтовано вибір технічних і програмних засобів реалізації, розроблено загальну архітектуру системи, структуру бази даних, механізм перевірки реквізитів і модуль генерації документів на основі шаблонів.

У результаті реалізації кваліфікаційної роботи створено програмний засіб, який забезпечує автоматизоване формування типових договорів, перевірку правильності введених даних та збереження сформованих документів.






Підпис здобувача

30.05.2026

Дата

ЗМІСТ

Вступ.....	4
1 Аналіз предметної області та постановка задачі	6
1.1 Загальна характеристика процесу підготовки та оформлення типових договорів	6
1.2 Поняття та структура типового договору в інформаційних системах	8
1.3 Аналіз існуючих підходів до формування та обробки договорів	11
1.4 Проблеми ручного та напівавтоматизованого формування договорів....	15
1.5 Вимоги до системи автоматизованої генерації та перевірки договорів	17
2 Проекування інформаційної системи автоматизованої генерації та перевірки договорів.....	20
2.1 Загальна архітектура інформаційної системи	20
2.2 Обґрунтування вибору технічних та програмних засобів реалізації.....	23
2.3 Розроблення структури даних та моделі інформаційної системи.....	27
2.4 Організація обробки даних та алгоритму генерації і перевірки договорів	29
2.5 Організація взаємодії клієнтської та серверної частин	33
2.6 Особливості забезпечення коректності даних та безпеки системи	36
3 Реалізація інформаційної системи автоматизованої генерації та перевірки договорів.....	39
3.1 Реалізація серверної частини системи	39
3.2 Реалізація структури бази даних та зберігання інформації	43
3.3 Реалізація механізму перевірки введених реквізитів	47
3.4 Реалізація генерації договорів на основі шаблонів	52
3.5 Реалізація клієнтського інтерфейсу користувача	57
Висновки	63
Перелік джерел посилань	65

КВРІСТ.230156.23.01.05 ПЗ									
Зм.	Арк.	№докум.	Підпис	Дата	Система автоматизованої генерації та перевірки типових договорів на основі введених реквізитів та шаблонів. Пояснювальна записка	Літера	Аркуш	Аркушів	
Виконав		МИХАЙЛОВ				у		2	64
Перевір.		Олег ВОЙЧУР							
Н.контр.		Тетяна КИСІЛЬ							
Затвер.		Ольга ПАВЛОВА		01/06				ХНУ ICTc-23-1	

Додаток А Архітектура пз проекту	71
Додаток Б Архітектура інформаційної системи	72
Додаток В Алгоритм роботи системи	73

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

В умовах активної цифровізації ділових процесів питання підготовки та оформлення договірної документації набуває особливої ваги. Договори залишаються основним інструментом фіксації прав та обов'язків сторін у більшості сфер діяльності, починаючи від комерційних відносин і завершуючи наданням послуг та виконанням робіт. Попри широке використання комп'ютерних засобів, на практиці значна частина операцій, пов'язаних зі створенням договорів, усе ще виконується вручну або з використанням обмежено автоматизованих інструментів. Це призводить до суттєвих витрат часу, підвищує ймовірність появи помилок у реквізитах і ускладнює контроль узгодженості тексту документа на різних етапах його підготовки.

Сучасні інформаційні системи поступово змінюють підхід до роботи з документами, перетворюючи договір із простого текстового файлу на повноцінний елемент цифрового середовища. У такому підході документ пов'язується з іншими даними організації, зокрема відомостями про контрагентів, фінансовими показниками та внутрішніми процесами. Це дозволяє розглядати підготовку договорів не як ізольовану операцію, а як частину більш широкого інформаційного процесу, у межах якого важливу роль відіграють стандартизація, узгодженість і контроль якості даних. За таких умов особливого значення набуває використання шаблонів, структурованих реквізитів і формалізованих правил їх перевірки.

Практика показує, що навіть за наявності шаблонів договорів значна частина реквізитів вводиться вручну, що створює умови для появи неточностей, дублювання інформації та логічних неузгодженостей між окремими частинами документа. Виправлення подібних помилок зазвичай відбувається вже на етапі погодження або перевірки договору, що збільшує тривалість підготовки та знижує загальну ефективність роботи з документацією. Унаслідок цього зростає потреба у таких програмних рішеннях, які не лише автоматизують процес

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

формування тексту договору, а й забезпечують перевірку коректності введених даних ще на ранніх етапах роботи.

Окремої уваги потребує питання узгодженості та актуальності даних, що використовуються під час підготовки договорів. У багатьох організаціях інформація про контрагентів, умови співпраці та фінансові показники зберігається в різних джерелах, і під час ручного перенесення цих відомостей у текст документа зростає ризик появи розбіжностей. Автоматизований підхід до генерації договорів дозволяє зменшити кількість подібних ситуацій за рахунок використання єдиних структур даних і централізованого керування шаблонами та реквізитами. Це, у свою чергу, сприяє підвищенню цілісності інформації та зменшенню залежності результату від людського фактора.

Не менш важливим є й аспект інформаційної безпеки, оскільки договори часто містять чутливі відомості, пов'язані з фінансами та персональними даними. У межах цифрового середовища обробка таких документів потребує не лише зручних інструментів генерації, а й продуманих механізмів контролю доступу та захисту інформації. Це ще раз підкреслює, що сучасна система роботи з договорами має поєднувати функції створення документів, перевірки коректності даних і забезпечення належного рівня безпеки.

Розглянуто підходи до побудови системи автоматизованої генерації та перевірки типових договорів на основі введених реквізитів та шаблонів. У процесі виконання роботи проаналізовано особливості підготовки договірної документації в сучасних умовах, визначено ключові проблеми ручних і напіваавтоматизованих підходів та сформульовано вимоги до програмного рішення, здатного усунути виявлені недоліки. Запропонована система орієнтується на поєднання шаблонного підходу з використанням структурованих даних і механізмів автоматизованої перевірки, що створює передумови для підвищення ефективності, надійності та зручності роботи з договірною документацією в сучасних інформаційних системах.

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Загальна характеристика процесу підготовки та оформлення типових договорів

У межах бакалаврської кваліфікаційної роботи процес підготовки та оформлення типових договорів розглядається як послідовність взаємопов'язаних дій, спрямованих на перетворення шаблонного тексту на юридично коректний та змістовно заповнений документ. У сучасних умовах цифровізації ділових процесів договір перестає бути виключно статичним текстовим файлом і все частіше розглядається як елемент інформаційної системи, що проходить певні етапи життєвого циклу: створення, редагування, погодження, збереження та використання [31, 32]. У межах таких підходів значну роль відіграють засоби автоматизації, які дозволяють уніфікувати структуру документів і зменшити залежність від ручної праці.

Типовий договір, як правило, формується на основі заздалегідь підготовленого шаблону, у якому визначено сталі фрагменти тексту та спеціальні місця для підстановки змінних даних. До таких даних належать реквізити сторін, дати укладання, суми, адреси, ідентифікаційні коди, банківські реквізити та інші параметри, без яких документ не може вважатися завершеним. У сучасних програмних системах цей підхід доповнюється використанням формалізованих описів структур даних, що застосовуються для обміну інформацією між компонентами системи та для перевірки коректності введених значень, зокрема через специфікації OpenAPI та механізми валідації на основі JSON Schema [10]. Це дозволяє не лише уніфікувати представлення реквізитів, а й закласти основу для їх автоматизованої перевірки.

Процес підготовки договору в реальних умовах часто включає кілька етапів, починаючи від вибору відповідного шаблону і завершуючи формуванням фінального файлу у потрібному форматі. На кожному з цих етапів можливе виникнення помилок, пов'язаних із неповнотою даних, некоректним форматом

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

окремих реквізитів або неузгодженістю між різними частинами документа. Саме тому у сучасних системах електронного документообігу значна увага приділяється питанням автоматизованої перевірки введеної інформації та контролю її відповідності заданим правилам [5]. Відповідно, вдається зменшити кількість повторних виправлень і підвищити загальну надійність процесу підготовки документів.

Окремим аспектом є питання інформаційної безпеки та захисту даних, що обробляються під час формування договорів. Реквізити сторін і фінансові показники належать до чутливої інформації, тому сучасні підходи до побудови інформаційних систем орієнтуються на вимоги міжнародних стандартів і рекомендацій у сфері захисту інформації та безпечної розробки програмного забезпечення [1, 2]. У межах цього підходу процес підготовки договору розглядається не лише як операція з текстом, а як складова більш широкого інформаційного процесу, у якому враховуються питання цілісності, конфіденційності та коректності даних.

Розвиток електронного документообігу та цифрових сервісів також впливає на зміну підходів до укладання та оформлення договорів. На міжнародному рівні питання автоматизованого контрагування та використання програмних систем для формування правових документів відображено у матеріалах UNCITRAL, де визначено загальні принципи застосування автоматизованих засобів у договірних відносинах [18, 20]. На рівні Європейського Союзу та України розвиток електронної ідентифікації та довірчих послуг створює додаткові передумови для широкого використання електронних форм договорів у практичній діяльності [11, 17]. Унаслідок цього процес підготовки типових договорів поступово інтегрується у загальну інфраструктуру цифрових сервісів і інформаційних систем.

З технічної точки зору оформлення договорів у сучасних умовах часто здійснюється із застосуванням спеціалізованих бібліотек та інструментів для роботи з документами, які підтримують генерацію файлів у поширених

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

форматах, зокрема DOCX або PDF [44, 45]. Використання таких засобів дозволяє поєднати шаблонний підхід із програмною логікою обробки даних та забезпечити формування документів у стандартизованому вигляді. При цьому сам процес підготовки договору дедалі більше набуває рис формалізованої процедури, у межах якої кожен крок від введення реквізитів до отримання фінального файлу може бути описаний, автоматизований і проконтрольований.

Спираючись на розглянуті аспекти, процес підготовки та оформлення типових договорів у сучасних інформаційних системах постає як поєднання роботи з шаблонами, структурованими даними та механізмами перевірки коректності, що забезпечує узгодженість документів, зменшує кількість помилок і підвищує ефективність роботи з договірною документацією [31, 34]. Це створює необхідне підґрунтя для подальшого проектування системи автоматизованої генерації та перевірки договорів у межах цієї бакалаврської кваліфікаційної роботи.

1.2 Поняття та структура типового договору в інформаційних системах

У сучасних умовах цифровізації ділових процесів типовий договір розглядається як формалізований документ, що має сталу логічну структуру та набір змінних реквізитів, які визначають конкретні умови правочину. В інформаційних системах такий документ перестає бути лише текстовим файлом і набуває властивостей структурованого інформаційного об'єкта, для якого визначено правила створення, заповнення, перевірки та подальшого використання у процесах керування документацією. Це дозволяє уніфікувати подання договорів, зменшити кількість помилок під час їх підготовки та забезпечити узгодженість змісту в межах організаційних процедур.

Поняття типового договору тісно пов'язується з використанням шаблонів, у яких зафіксовано сталі фрагменти тексту та спеціальні маркери для підстановки змінних даних. До таких даних належать відомості про сторони,

					КВРІСТ.230156.23.01.05 ПЗ	Арк. 8
Зм.	Арк.	№ докум.	Підпис	Дата		

предмет договору, строки виконання зобов'язань, фінансові показники, адреси та інші реквізити, без яких документ не може вважатися повноцінним. У цифрових системах ці реквізити подаються у вигляді структурованих полів, що описуються через формальні моделі даних та інтерфейси взаємодії між програмними компонентами [8]. Застосування схем даних і формальних описів, зокрема на основі JSON Schema, дає змогу однозначно визначити формат кожного поля та правила його заповнення, створюючи підґрунтя для автоматизованої обробки та перевірки інформації [10].

З погляду внутрішньої організації документа структура типового договору в цифровому середовищі має ієрархічний характер. На верхньому рівні формується загальний каркас, який включає назву, преамбулу, основні розділи та заключні положення, що залишаються сталими для певного виду договорів. Наступний рівень охоплює змінні фрагменти, у які підставляються конкретні значення реквізитів, отримані з форм введення або з інших інформаційних джерел. Окрему роль відіграють метадані, що застосовуються для ідентифікації документа, керування версіями, пошуку та інтеграції у системи керування життєвим циклом договорів. Такий поділ спрощує як генерацію документа, так і його подальше використання в автоматизованих інформаційних процесах.

Важливим елементом структури типового договору є правила валідації реквізитів. Для кожного поля можуть задаватися вимоги щодо обов'язковості заповнення, формату, допустимого діапазону значень або логічних зв'язків з іншими полями. Реалізація таких правил дозволяє автоматично виявляти помилки ще на етапі введення даних і запобігати формуванню некоректних документів [5]. У сучасних програмних рішеннях ці механізми тісно пов'язані з формальними описами схем даних та сервісними інтерфейсами, що забезпечує узгодженість перевірок на різних етапах обробки інформації [10].

У контексті інтеграції з корпоративними інформаційними системами типовий договір розглядається не як ізольований файл, а як елемент єдиного інформаційного простору, пов'язаний із довідниками контрагентів, фінансовими

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 9
Зм.	Арк.	№ докум.	Підпис	Дата		

підсистемами та електронними архівами документів. Це означає, що структура договору має враховувати не лише вимоги до текстового оформлення, а й потреби збереження, пошуку, аналізу та подальшого використання даних у суміжних бізнес-процесах.

З урахуванням вимог інформаційної безпеки структура типового договору в цифрових системах доповнюється засобами контролю доступу та захисту даних. Обробка реквізитів сторін і фінансових показників здійснюється відповідно до сучасних підходів до забезпечення конфіденційності, цілісності та доступності інформації, що відображено у міжнародних стандартах і рекомендаціях у сфері інформаційної безпеки та безпечної розробки програмного забезпечення [1, 2]. Унаслідок цього структура документа формується з урахуванням не лише зручності генерації, а й вимог до надійності та захищеності даних.

З технічної точки зору типовий договір у межах інформаційної системи реалізується у вигляді шаблону, пов'язаного з механізмами генерації електронних документів у поширених форматах, таких як DOCX або PDF. Це дає змогу поєднати зручність роботи з документами для користувачів із можливістю автоматизованого формування файлів на основі структурованих даних. При цьому структура договору зберігає чіткий поділ між сталими та змінними елементами, що спрощує підтримку та розширення набору шаблонів.

Зважаючи на наведені аргументи, поняття та структура типового договору в інформаційних системах постають як поєднання шаблонного тексту, формалізованого набору реквізитів, правил їх перевірки та допоміжних метаданих, які забезпечують інтеграцію документа у цифрові процеси керування інформацією та підтримують автоматизовану генерацію і контроль коректності змісту [31].

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

1.3 Аналіз існуючих підходів до формування та обробки договорів

Найпоширенішими залишаються системи керування життєвим циклом договорів, у межах яких об'єднуються засоби створення, зберігання, погодження та архівування документів. До таких рішень належать, зокрема, платформи класу Contract Lifecycle Management, що використовуються у корпоративному середовищі та орієнтовані на централізовану роботу з шаблонами й готовими договорами.

Одним із прикладів такого підходу є рішення DocuSign CLM, яке поєднує засоби керування шаблонами договорів, автоматизоване заповнення реквізитів та підтримку електронного підпису, інтерфейс зображено на рисунку 1.1. Застосування подібних платформ дозволяє скоротити час підготовки документів і стандартизувати процеси роботи з контрактами, однак водночас такі системи зазвичай мають складну архітектуру та орієнтовані передусім на потреби великих організацій [31].

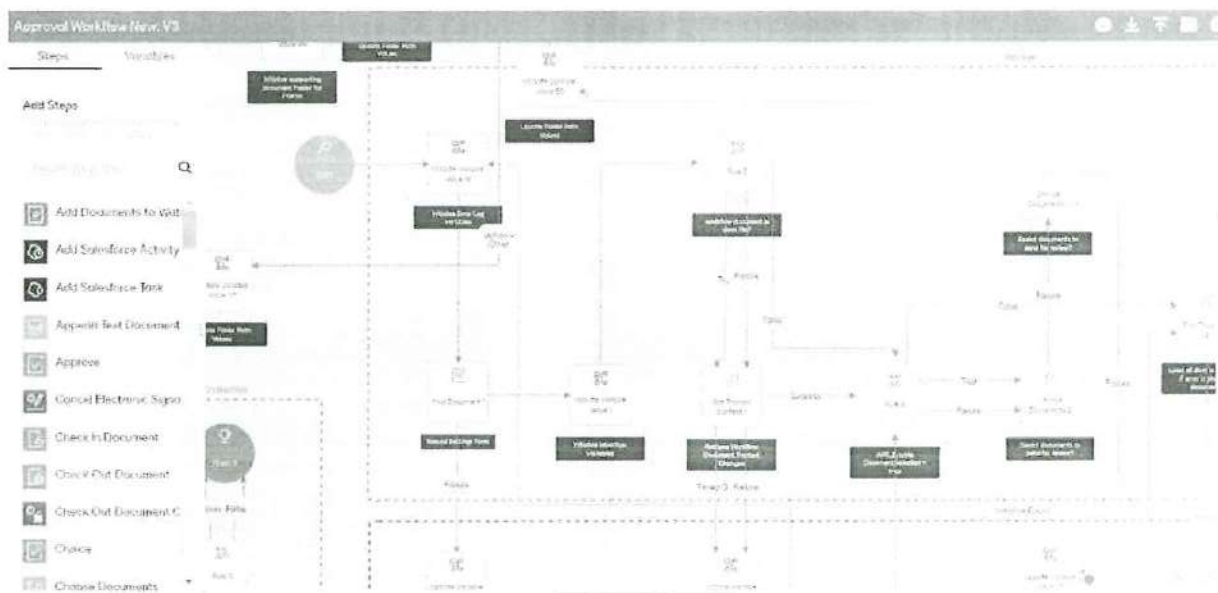


Рисунок 1.1 – Інтерфейс DocuSign CLM [51]

Аналогічну нішу займають корпоративні рішення, описані у звітах World Commerce & Contracting та KPMG, де формування договорів розглядається як

частина ширшої екосистеми бізнес-процесів і корпоративних інформаційних систем, інтерфейс зображено на рисунку 1.2 [32, 34].

Схожий підхід реалізовано також у таких платформах, як Ironclad та ContractWorks, які орієнтовані на автоматизацію процесів створення, погодження та зберігання договорів у межах корпоративного середовища. Ці системи надають інструменти для роботи з шаблонами, підтримують централізоване керування документами та дозволяють зменшити обсяг ручних операцій під час підготовки договорів. Водночас їх функціональність зазвичай зосереджена на організації документообігу та управлінні життєвим циклом контрактів, тоді як можливості глибокої перевірки логічної узгодженості реквізитів часто обмежуються базовими правилами валідації.



Рисунок 1.2 – ContractWorks [52]

Ще нішу займають рішення на кшталт PandaDoc та Adobe Acrobat Sign, які поєднують засоби створення документів, роботу з шаблонами та підтримку електронного підпису інтерфейс PandaDoc зображено на рисунку 1.3. Такі

платформи зручні для швидкого формування типових договорів і комерційних пропозицій, однак у більшості випадків вони орієнтовані передусім на підготовку та підписання документів, а не на комплексну перевірку внутрішньої узгодженості реквізитів і структурних залежностей між даними в документі.

Крім комерційних платформ, на практиці використовуються й спеціалізовані бібліотеки та інструменти для програмної генерації документів, зокрема засоби для формування файлів у форматах DOCX і PDF на основі шаблонів. Такі підходи широко застосовуються у власних програмних розробках, оскільки дозволяють гнучко інтегрувати механізм створення документів у бізнес-логіку інформаційних систем. Водночас у подібних рішеннях питання перевірки коректності реквізитів і логічної цілісності договору зазвичай потребують окремої реалізації на рівні прикладної логіки.

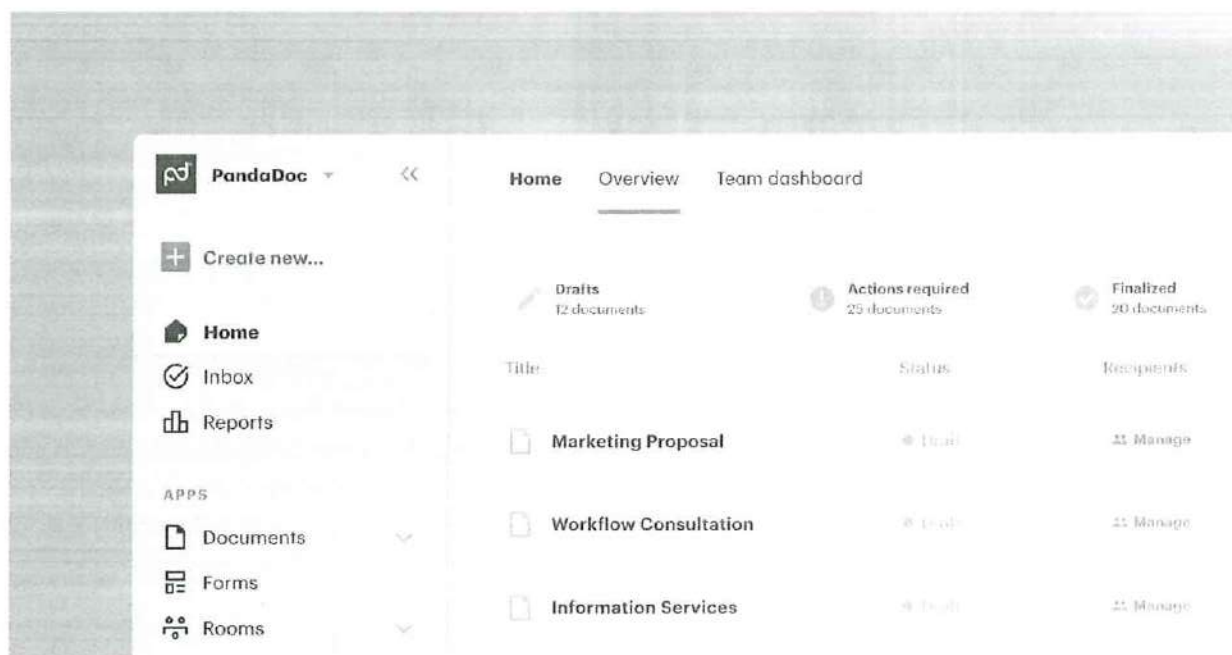


Рисунок 1.3 – PandaDoc [53]

Окрему групу становлять комерційні платформи, що зосереджуються переважно на зручності підготовки та керування документами без глибокої інтеграції з іншими системами. У таких рішеннях зазвичай реалізовано роботу з

шаблонами, базові механізми заповнення полів та збереження документів, а також елементарні інструменти пошуку та категоризації. Аналітичні огляди ринку показують, що подібні системи широко застосовуються у малому та середньому бізнесі завдяки відносній простоті впровадження, однак їхні можливості щодо перевірки коректності реквізитів і складної логіки обробки даних часто залишаються обмеженими [33].

Поряд із спеціалізованими платформами на практиці продовжують використовуватися універсальні офісні інструменти на основі текстових шаблонів, де договори формуються шляхом ручного або напівавтоматизованого заповнення полів. Такий підхід зберігає популярність через низький поріг входження, однак він не забезпечує належного рівня контролю коректності даних і не підтримує централізоване керування структурою документів. У результаті якість підготовлених договорів значною мірою залежить від людського фактора, а процес перевірки потребує додаткових витрат часу.

З технічної точки зору значна частина сучасних рішень спирається на використання бібліотек і сервісів для генерації документів у форматах DOCX або PDF, що дозволяє відокремити логіку підготовки даних від представлення фінального документа. Такі інструменти активно використовуються як у комерційних продуктах, так і в індивідуальних програмних розробках, проте самі по собі вони не вирішують питання комплексної перевірки реквізитів і логічної узгодженості змісту договорів.

Окремим напрямом розвитку є застосування інтелектуальних інструментів аналізу та обробки юридичних текстів, які використовуються для порівняння договорів, витягування положень та підтримки підготовки документів [24]. Такі підходи демонструють перспективи подальшої автоматизації, однак у практичних системах вони переважно доповнюють, а не замінюють класичні механізми шаблонної генерації документів.

З огляду на вищезазначене, аналіз існуючих рішень показує, що на сьогодні використовуються як потужні корпоративні платформи керування договорами,

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

так і спрощені інструменти на основі шаблонів та бібліотек генерації документів. Кожен із цих підходів має власні переваги та обмеження з точки зору складності впровадження, рівня автоматизації та можливостей перевірки коректності реквізитів [33]. Це підтверджує доцільність розроблення спеціалізованої системи, у якій процеси автоматизованої генерації та перевірки типових договорів поєднуються в єдиному програмному рішенні.

1.4 Проблеми ручного та напівавтоматизованого формування договорів

Попри активне впровадження цифрових інструментів у сфері документообігу, у багатьох організаціях процес підготовки договорів і надалі ґрунтується на ручному або напівавтоматизованому заповненні шаблонів. За рахунок цього вдається передбачати використання текстових редакторів або простих шаблонних форм, у межах яких значна частина реквізитів вводиться вручну. Аналітичні огляди та звіти, присвячені керуванню договорами, свідчать про те, що істотна частка робочого часу витрачається саме на рутинні операції з підготовки та повторної перевірки документів, а не на їх змістовне опрацювання [31, 33].

Однією з основних проблем ручного підходу залишається висока ймовірність помилок у реквізитах сторін, фінансових показниках та датах. За відсутності формалізованих механізмів перевірки даних помилки можуть накопичуватися та виявлятися лише на пізніх етапах погодження документа, що призводить до додаткових витрат часу та необхідності повторного редагування [27, 41]. Сучасні підходи до побудови інформаційних систем наголошують на доцільності використання формальних схем даних і правил валідації, однак у ручних процесах ці можливості практично не застосовуються [9, 10].

Напівавтоматизовані інструменти частково зменшують обсяг ручної роботи за рахунок використання полів для підстановки значень і базових перевірок формату, проте вони зазвичай не забезпечують контролю логічної

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

узгодженості між різними частинами документа. У результаті можливі ситуації, коли окремі реквізити формально відповідають заданим вимогам, але суперечать один одному за змістом або контекстом договору [22, 45]. Подібні обмеження знижують загальну надійність напівавтоматизованих рішень і зберігають залежність від людського фактора.

Окремою проблемою є підтримка актуальності шаблонів і єдиного стилю оформлення договорів. За відсутності централізованого керування шаблонами різні виконавці можуть використовувати різні версії документів, що призводить до появи застарілих формулювань або розбіжностей у структурі договорів [14, 38]. У таких умовах ускладнюється контроль змін і забезпечення єдиного підходу до оформлення договірної документації.

Суттєвим обмеженням ручних і напівавтоматизованих підходів є також слабка інтеграція з іншими інформаційними системами. Дані про контрагентів, фінансові показники або об'єкти договорів часто зберігаються в окремих джерелах, однак під час підготовки документів вони переносяться вручну. Це не лише збільшує трудомісткість процесу, а й підвищує ризик розбіжностей між первинними даними та текстом договору [18, 29]. Відсутність уніфікованих інтерфейсів обміну інформацією ускладнює побудову цілісного інформаційного середовища.

З точки зору інформаційної безпеки ручна обробка договорів також створює додаткові ризики. Копіювання файлів і передавання їх через неформалізовані канали ускладнює контроль доступу та аудит дій користувачів, а також підвищує ймовірність несанкціонованого поширення чутливої інформації [3, 40]. Сучасні стандарти та рекомендації у сфері захисту інформації орієнтують на використання централізованих механізмів контролю доступу та забезпечення цілісності даних, що важко реалізувати в умовах фрагментованих ручних процесів [11, 24].

Окремою проблемою залишається масштабування ручних і напівавтоматизованих підходів у разі зростання обсягів договірної документації.

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 16
Зм.	Арк.	№ докум.	Підпис	Дата		

Зі збільшенням кількості договорів зростають витрати часу на їх підготовку, перевірку та супровід, а також підвищується навантаження на персонал, відповідальний за контроль якості документів [34, 50]. У таких умовах ефективність ручних процедур швидко знижується, а кількість помилок має тенденцію до зростання.

Крім того, використання лише засобів генерації файлів у форматах DOCX або PDF без комплексної логіки перевірки реквізитів не вирішує проблеми якості підготовки договорів, оскільки такі інструменти зосереджуються переважно на представленні документа, а не на контролі його змісту [44, 46]. Це ще раз підкреслює обмеженість напівавтоматизованих підходів, у яких відсутня єдина система правил перевірки даних.

Спираючись на розглянуті аспекти, ручні та напівавтоматизовані підходи до формування договорів характеризуються високою залежністю від людського фактора, обмеженими можливостями валідації даних, слабкою інтеграцією з іншими інформаційними системами та підвищеними ризиками з точки зору безпеки й масштабованості [34, 50]. Це підтверджує доцільність переходу до спеціалізованих засобів автоматизованої генерації та перевірки типових договорів, у яких зазначені обмеження можуть бути враховані на рівні архітектури системи.

1.5 Вимоги до системи автоматизованої генерації та перевірки договорів

Сучасні умови цифровізації документообігу формують потребу у створенні спеціалізованих програмних рішень, здатних забезпечити не лише швидке формування договорів, а й контроль коректності введених даних та узгодженості змісту документа. Практика використання систем керування договорами та засобів електронного документообігу свідчить про те, що ефективність таких рішень значною мірою залежить від правильно сформульованих вимог до їх функціональності та архітектури [31]. У цьому

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

контексті система автоматизованої генерації та перевірки договорів має розглядатися як складова загального інформаційного середовища організації.

Передусім доцільно виділити вимоги, пов'язані з підтримкою шаблонів договорів і структурованих реквізитів. Система повинна забезпечувати зберігання та використання уніфікованих шаблонів документів, у яких чітко відокремлено сталі та змінні фрагменти тексту. Для реквізитів мають застосовуватися формалізовані структури даних, що дозволяє уніфікувати їх подання та спростити подальшу обробку [8]. Використання схем даних і формальних описів полів створює підґрунтя для автоматизованої перевірки коректності значень і зменшення кількості помилок на етапі введення інформації [10].

Важливою вимогою є наявність розвинених механізмів валідації реквізитів і логічного контролю узгодженості даних. Перевірка має охоплювати не лише формат окремих полів, а й взаємозв'язки між ними, наприклад, відповідність дат, сум і умов договору. Запропоноване рішення відповідає сучасним рекомендаціям щодо перевірки вхідних даних у програмних системах і зменшення ризиків помилок, пов'язаних із людським фактором [22]. Реалізація комплексної валідації дозволяє підвищити надійність підготовлених документів і скоротити кількість повторних виправлень.

Окрему групу вимог становлять вимоги до інтеграції системи з іншими інформаційними ресурсами. У сучасних організаціях дані про контрагентів, фінансові показники та об'єкти договорів часто зберігаються в різних підсистемах, тому система генерації договорів має підтримувати обмін інформацією через стандартизовані інтерфейси [9]. Це дозволяє зменшити обсяг ручного введення даних і підвищити цілісність інформаційного простору, а також забезпечити повторне використання наявних даних у різних бізнес-процесах [29].

Суттєве значення мають вимоги до безпеки обробки інформації та контролю доступу. Оскільки договори містять чутливі дані, система повинна

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

підтримувати механізми автентифікації користувачів, розмежування прав доступу та журналювання дій. Сучасні стандарти й рекомендації у сфері інформаційної безпеки орієнтують на впровадження комплексних заходів захисту даних, що включають забезпечення конфіденційності, цілісності та доступності інформації. Урахування цих вимог є необхідною умовою для використання системи в реальних умовах експлуатації.

З технічної точки зору важливою є підтримка генерації документів у поширених форматах електронних файлів, зокрема DOCX і PDF, які широко застосовуються в діловому середовищі. Це дозволяє забезпечити сумісність із наявними засобами перегляду та зберігання документів, а також спростити інтеграцію з іншими інформаційними системами. Водночас механізм генерації файлів має бути тісно пов'язаний із логікою перевірки реквізитів, щоб формування фінального документа відбувалося лише після підтвердження коректності даних [41].

Не менш важливою є вимога до зручності використання та масштабованості системи. Інтерфейс має забезпечувати зрозумілий процес введення реквізитів, вибору шаблонів і отримання готового документа, що зменшує навантаження на користувачів і підвищує продуктивність роботи [33]. Водночас архітектура системи повинна передбачати можливість розширення функціональності, додавання нових типів договорів і збільшення обсягів оброблюваних даних без суттєвих змін базових компонентів [34].

Узагальнюючи, вимоги до системи автоматизованої генерації та перевірки договорів охоплюють підтримку шаблонів і структурованих реквізитів, реалізацію комплексної валідації даних, інтеграцію з іншими інформаційними системами, забезпечення інформаційної безпеки, підтримку поширених форматів документів і зручність використання з можливістю масштабування. Дотримання цих вимог створює основу для побудови надійного та ефективного програмного рішення, здатного автоматизувати процес підготовки типових договорів і підвищити якість роботи з договірною документацією.

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОМАТИЗОВАНОЇ ГЕНЕРАЦІЇ ТА ПЕРЕВІРКИ ДОГОВОРІВ

2.1 Загальна архітектура інформаційної системи

У цьому підрозділі розглядається загальна архітектура інформаційної системи автоматизованої генерації та перевірки типових договорів, яка сформована з урахуванням необхідності забезпечення повного циклу обробки документів від введення реквізитів до формування готового файлу. Архітектурне рішення орієнтоване на побудову гнучкої, масштабованої та логічно узгодженої системи, здатної працювати в умовах реального використання та обробки значних обсягів даних.

Основою запропонованої системи виступає клієнт-серверна архітектура, яка забезпечує розділення функціональності між інтерфейсною частиною та серверною логікою. Це дає змогу ізолювати процеси обробки даних від процесів взаємодії з користувачем, що підвищує надійність системи та спрощує її подальше розширення. Клієнтська частина відповідає за введення реквізитів, вибір шаблонів договорів і відображення результатів роботи системи, тоді як серверна частина реалізує основну бізнес-логіку, пов'язану з обробкою даних, перевіркою їх коректності та генерацією документів.

Клієнтський рівень системи реалізує інтерфейс взаємодії з користувачем, у межах якого забезпечується введення необхідних реквізитів договору, вибір типу документа та ініціація процесу генерації. Інтерфейс побудовано таким чином, щоб мінімізувати кількість помилок під час введення даних за рахунок використання структурованих форм, підказок і базових перевірок значень. На цьому рівні також відбувається первинна обробка даних, що дозволяє зменшити навантаження на серверну частину та прискорити реакцію системи на дії користувача.

Серверна частина інформаційної системи виконує ключову роль у реалізації функціональності. Вона забезпечує приймання запитів від клієнта,

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

обробку введених реквізитів, виконання перевірки даних та формування фінального документа. Архітектурно серверна частина поділяється на кілька логічних модулів, кожен з яких відповідає за окремий етап обробки інформації. До основних модулів належать модуль обробки запитів, модуль валідації даних, модуль роботи з шаблонами та модуль генерації документів.

Модуль обробки запитів відповідає за приймання даних від клієнтської частини, їх первинну обробку та маршрутизацію до відповідних компонентів системи. У цьому модулі реалізується логіка взаємодії між різними частинами серверної архітектури, що дозволяє забезпечити узгоджену роботу системи. Особливістю реалізації є можливість обробки кількох запитів одночасно, що є важливим у випадку багатокористувацького режиму роботи.

Модуль валідації даних виконує перевірку коректності введених реквізитів відповідно до заданих правил. У цьому модулі реалізовано як перевірку формату окремих полів, так і логічний контроль узгодженості даних між собою. Наприклад, перевіряється відповідність дат, правильність числових значень, обов'язковість заповнення полів та взаємозв'язки між різними параметрами договору. Це дозволяє виявляти помилки ще до етапу генерації документа, що суттєво підвищує якість кінцевого результату.

Модуль роботи з шаблонами забезпечує зберігання, обробку та використання шаблонів договорів. У цьому модулі реалізовано механізм підстановки значень у відповідні місця шаблону, що дозволяє автоматично формувати текст документа на основі введених реквізитів. Шаплони зберігаються у структурованому вигляді, що спрощує їх редагування, оновлення та повторне використання. Це дає змогу забезпечувати уніфікований підхід до формування договорів різних типів.

Модуль генерації документів відповідає за формування фінального файлу у вибраному форматі. На цьому етапі відбувається об'єднання шаблонного тексту з перевіреними реквізитами та створення готового документа, придатного для подальшого використання або збереження. Реалізація цього модуля

передбачає підтримку поширених форматів документів, що забезпечує сумісність із зовнішніми системами та програмними засобами.

Окреме місце в архітектурі займає підсистема зберігання даних, яка відповідає за збереження шаблонів, реквізитів, згенерованих документів і службової інформації. Використання централізованого сховища дозволяє забезпечити цілісність даних, спростити їх пошук і організувати ефективну роботу з історією документів. Крім того, це створює можливість повторного використання даних і автоматизації процесів, пов'язаних із підготовкою договорів. Структурну схему роботи подано на рисунку 2.1.

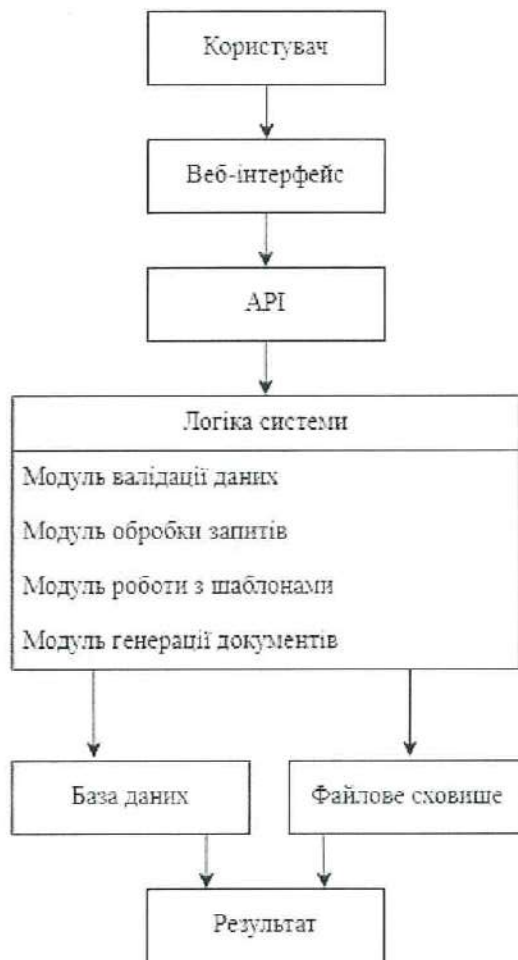


Рисунок 2.1 – Структурна схема

Взаємодія між клієнтською та серверною частинами здійснюється через програмний інтерфейс, який забезпечує передавання даних у структурованому

вигляді. Використання цієї технології допомагає відокремити інтерфейс від бізнес-логіки та забезпечити гнучкість системи при зміні окремих компонентів. Обмін даними організовано таким чином, щоб забезпечити швидкість обробки запитів і надійність передавання інформації.

У межах архітектури також враховано питання безпеки та контролю доступу до даних. Система передбачає механізми автентифікації користувачів, розмежування прав доступу та захисту інформації від несанкціонованого використання. Це особливо важливо з огляду на те, що договори містять чутливу інформацію, яка потребує належного рівня захисту.

На основі проведеного аналізу можна стверджувати, що загальна архітектура інформаційної системи сформована як багаторівнева структура, у якій кожен компонент виконує чітко визначену функцію, а їх взаємодія забезпечує повний цикл автоматизованої генерації та перевірки договорів. Це допомагає досягти високого рівня надійності, гнучкості та ефективності системи, а також створює основу для подальшого розширення її функціональності та адаптації до різних умов використання.

2.2 Обґрунтування вибору технічних та програмних засобів реалізації

Під час розроблення інформаційної системи автоматизованої генерації та перевірки типових договорів особливу увагу приділено вибору технічних і програмних засобів, оскільки саме від цього залежить ефективність функціонування системи, її масштабованість, зручність використання та можливість подальшого розширення. Обраний стек технологій сформовано з урахуванням специфіки задачі, що передбачає обробку текстових даних, роботу зі структурованими реквізитами та генерацію документів у поширених форматах.

Як основну мову програмування доцільно використати Python, що зумовлено його універсальністю, широкою підтримкою бібліотек для обробки

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 23
Зм.	Арк.	№ докум.	Підпис	Дата		

даних і документів, а також зручністю реалізації серверної логіки. Використання Python дозволяє швидко реалізувати основні функціональні можливості системи, пов'язані з валідацією даних, роботою з шаблонами та генерацією текстових документів. Крім того, ця мова добре підходить для побудови серверних застосунків і має розвинену екосистему інструментів для інтеграції з веб-технологіями.

Для реалізації серверної частини системи обрано фреймворк FastAPI, який забезпечує високу швидкодію, підтримку асинхронної обробки запитів та зручну організацію програмного інтерфейсу. Використання цього фреймворку дозволяє реалізувати чітко структуровану серверну архітектуру, у межах якої кожен компонент відповідає за окрему функцію. Додатковою перевагою є автоматична генерація документації для API, що спрощує тестування та подальший розвиток системи.

Для організації зберігання даних обрано реляційну базу даних, зокрема SQLite або PostgreSQL, залежно від умов розгортання системи. SQLite доцільно використовувати на етапі розроблення та тестування, оскільки вона не потребує окремого серверного розгортання та забезпечує достатній рівень функціональності для невеликих обсягів даних. PostgreSQL, у свою чергу, дозволяє реалізувати більш масштабоване рішення, орієнтоване на багатокористувацький режим роботи та обробку значних обсягів інформації. Використання реляційної моделі забезпечує чітку організацію даних і підтримку зв'язків між сутностями, що є важливим для системи, яка працює з договорами та реквізитами.

Для реалізації механізмів валідації даних доцільно використати бібліотеку Pydantic, яка дозволяє описувати структури даних у вигляді моделей та автоматично перевіряти їх відповідність заданим правилам. Це відкриває можливість забезпечувати централізований контроль коректності введених реквізитів і дозволяє уникнути дублювання логіки перевірки в різних частинах системи. Використання моделей даних також спрощує взаємодію між

клієнтською та серверною частинами, оскільки забезпечує єдине представлення інформації.

Механізм роботи з шаблонами реалізовано із застосуванням бібліотеки Jinja2, яка дозволяє формувати текстові документи на основі шаблонів із використанням змінних і умовних конструкцій. Це забезпечує гнучкість у створенні різних типів договорів і дозволяє легко адаптувати шаблони до нових вимог без зміни основної логіки системи. Використання шаблонізатора дозволяє чітко відокремити структуру документа від даних, що підставляються, що є важливим для підтримки та розширення функціональності.

Для генерації документів у форматі DOCX доцільно використати бібліотеки python-docx або docxtempl, які забезпечують створення та заповнення документів на основі шаблонів. У випадку необхідності формування PDF-файлів може застосовуватися бібліотека WeasyPrint або ReportLab, що дозволяє створювати документи з високим рівнем форматування. Використання таких інструментів забезпечує сумісність системи з поширеними форматами документів і спрощує їх подальше використання.

Клієнтська частина системи може бути реалізована у вигляді веб-інтерфейсу із застосуванням сучасних технологій, таких як HTML, CSS та JavaScript, або фреймворків на кшталт React. Це забезпечує зручний доступ до системи через браузер і не потребує встановлення додаткового програмного забезпечення на стороні користувача. Інтерфейс організовано у вигляді форм для введення реквізитів, вибору шаблонів і відображення результатів генерації документів, що забезпечує інтуїтивно зрозумілу взаємодію з системою.

Обмін даними між клієнтською та серверною частинами організовано через HTTP-запити з використанням JSON-формату. Запропоноване рішення забезпечує уніфіковане представлення даних і спрощує інтеграцію системи з іншими інформаційними ресурсами. Крім того, використання стандартних протоколів і форматів дозволяє забезпечити гнучкість системи та можливість її подальшого розширення.

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

У межах вибору технічних засобів також враховано вимоги до безпеки та надійності системи. Реалізовано базові механізми захисту даних, зокрема перевірку введених значень, обмеження доступу до функціональних можливостей і контроль коректності обробки запитів. Це дозволяє забезпечити стабільну роботу системи та захист інформації від несанкціонованого використання. Обрані технології та їх призначення подано у таблиці 2.1.

Таблиця 2.1 – Обрані технології та їх призначення

Технологія	Призначення у системі	Обґрунтування вибору
Python	Реалізація серверної логіки	Обрано через простоту реалізації, велику кількість бібліотек для роботи з документами та зручність обробки даних
FastAPI	Побудова API та обробка запитів	Забезпечує високу швидкодію, підтримує асинхронність і дозволяє швидко створювати структуровані веб-сервіси
PostgreSQL / SQLite	Зберігання даних (реквізити, шаблони, документи)	PostgreSQL підходить для масштабованих рішень, SQLite – для простоти розгортання на етапі розробки
Pydantic	Валідація та структура даних	Дозволяє описувати моделі даних і автоматично перевіряти коректність введених значень
Jinja2	Робота з шаблонами договорів	Забезпечує гнучку підстановку даних у шаблон і підтримує умовну логіку формування тексту
docxtempl / python-docx	Генерація DOCX-документів	Дозволяє формувати документи на основі шаблонів із збереженням форматування
HTML / CSS / JavaScript	Реалізація клієнтського інтерфейсу	Забезпечує доступність системи через браузер і зручність взаємодії з користувачем
JSON	Формат обміну даними	Забезпечує уніфіковану передачу даних між клієнтом і сервером
HTTP/HTTPS	Передавання запитів і відповідей	Стандарт взаємодії у веб-системах, HTTPS додатково забезпечує захист даних

За результатами розробки обраний набір технічних і програмних засобів забезпечує реалізацію всіх необхідних функцій системи, включаючи обробку даних, перевірку реквізитів, генерацію документів і взаємодію з користувачем. Поєднання сучасних інструментів і технологій створює основу для побудови ефективного, гнучкого та масштабованого програмного рішення, яке відповідає поставленим задачам і може бути адаптоване до різних умов використання.

2.3 Розроблення структури даних та моделі інформаційної системи

У цьому підрозділі розглядається структура даних інформаційної системи автоматизованої генерації та перевірки типових договорів, а також логічна модель, яка забезпечує зберігання, обробку та взаємозв'язок усіх необхідних елементів. Коректна організація даних є одним із ключових аспектів побудови системи, оскільки саме вона визначає ефективність роботи з інформацією, швидкість доступу до неї та можливість подальшого розширення функціональності.

Інформаційна система орієнтована на роботу з кількома основними типами даних, серед яких центральне місце займають реквізити договорів, шаблони документів та результати їх генерації. Виходячи з цього, модель даних сформовано таким чином, щоб забезпечити чітке розмежування між структурою шаблону, введеними користувачем значеннями та фінальними документами. Завдяки цьому вдається уникнути дублювання інформації та спрощує процес її обробки.

Основною сутністю системи виступає «Договір», який у даному випадку розглядається не як окремий файл, а як набір пов'язаних даних, що включає посилання на шаблон, набір реквізитів і згенерований результат. Кожен договір формується на основі певного шаблону, який визначає структуру документа та місця для підстановки змінних значень. Таким чином, між сутностями «Договір»

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

і «Шаблон» встановлюється зв'язок типу «багато до одного», оскільки один шаблон може використовуватися для створення багатьох договорів.

Сутність «Шаблон» містить інформацію про структуру документа, включаючи текстові фрагменти та спеціальні маркери для підстановки значень. У системі шаблони зберігаються у вигляді окремих об'єктів, що дозволяє централізовано керувати їх змінами та повторно використовувати для формування нових договорів. Така реалізація забезпечує уніфікацію документів і спрощує їх підтримку.

Сутність «Реквізити» представляє набір даних, що вводяться користувачем і підставляються у відповідні місця шаблону. Ця сутність має складну структуру, оскільки включає різні типи даних: текстові значення, числові параметри, дати, а також логічні поля. Для кожного реквізиту визначаються правила перевірки, які дозволяють контролювати коректність введених значень. Реквізити пов'язані з конкретним договором, що забезпечує індивідуалізацію кожного створеного документа.

Окрему роль у моделі відіграє сутність «Користувач», яка дозволяє організувати роботу системи в багатокористувацькому режимі. Користувач пов'язується з договорами, які він створює або редагує, що дає змогу відстежувати історію змін і реалізувати механізми контролю доступу. Наявність цієї сутності також створює основу для подальшого розширення системи, наприклад, шляхом додавання ролей або прав доступу.

Сутність «Документ» використовується для зберігання результатів генерації договорів. Вона містить інформацію про сформований файл, включаючи його формат, шлях збереження та дату створення. Це відкриває можливість організувати зберігання готових документів і забезпечити швидкий доступ до них у разі потреби.

Взаємозв'язки між основними сутностями формують логічну модель інформаційної системи. Користувач створює договори, кожен договір пов'язаний із конкретним шаблоном і набором реквізитів, а результат генерації

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

зберігається у вигляді окремого документа. Така структура забезпечує чітке розмежування відповідальності між різними елементами системи та спрощує її масштабування.

З точки зору реалізації бази даних, структура моделі може бути представлена у вигляді реляційної схеми, де кожна сутність відповідає окремій таблиці. Таблиці пов'язані між собою за допомогою зовнішніх ключів, що дозволяє забезпечити цілісність даних і уникнути дублювання інформації. Використання реляційної моделі також спрощує виконання запитів і забезпечує ефективну роботу системи навіть при збільшенні обсягів даних.

Важливим аспектом розробки є нормалізація структури даних, що дозволяє зменшити надлишковість інформації та підвищити її узгодженість. Дані розподілено між таблицями таким чином, щоб кожен елемент зберігався лише в одному місці, а доступ до нього здійснювався через встановлені зв'язки. Це забезпечує зручність оновлення інформації та зменшує ризик виникнення помилок.

Отже, розроблена структура даних та модель інформаційної системи забезпечують ефективне зберігання, обробку та взаємозв'язок усіх елементів, необхідних для автоматизованої генерації та перевірки договорів. Запропонований підхід створює надійну основу для реалізації програмної частини системи та дозволяє забезпечити її гнучкість, масштабованість і відповідність поставленим задачам.

2.4 Організація обробки даних та алгоритму генерації і перевірки договорів

Організація обробки даних у системі автоматизованої генерації та перевірки типових договорів побудована як послідовний, чітко структурований процес, у якому кожен етап виконує визначену функцію та логічно пов'язаний із наступним. Цей метод дозволяє забезпечити не лише формування документа, а й контроль його коректності, узгодженості та відповідності заданим вимогам ще

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 29
Зм.	Арк.	№ докум.	Підпис	Дата		

до моменту отримання фінального результату. В основі реалізації закладено принцип поетапної трансформації даних: від первинного введення до формування завершеного документа.

Початковим етапом виступає введення реквізитів через клієнтський інтерфейс, де користувач заповнює структуровану форму, що відповідає обраному типу договору. Організація інтерфейсу побудована таким чином, щоб максимально спростити процес введення даних і водночас зменшити ймовірність помилок. Для цього застосовано розподіл полів за логічними групами, використання підказок, автоматичне форматування значень і базову перевірку на рівні інтерфейсу. Уже на цьому етапі здійснюється попередній контроль правильності введення, що дозволяє відсіяти частину помилок ще до передавання даних на сервер.

Після введення дані передаються до серверної частини системи, де розпочинається основний процес їх обробки. Першим кроком виконується приймання та структуризація даних відповідно до визначеної моделі. Це означає, що всі введені значення приводяться до єдиного формату та розподіляються за відповідними категоріями, що спрощує подальшу обробку. На цьому етапі також може виконуватися очищення даних від зайвих символів, приведення текстових значень до стандартного вигляду та усунення можливих неоднозначностей.

Наступним етапом виступає нормалізація даних, яка забезпечує їх уніфікацію. Наприклад, дати приводяться до єдиного формату, числові значення – до визначеного типу, а текстові дані – до узгодженого стилю. Нормалізація дозволяє уникнути ситуацій, коли одна й та сама інформація представлена в різних форматах, що може ускладнювати подальшу обробку або призводити до логічних помилок. Унаслідок цього формується єдиний узгоджений набір даних, придатний для використання в алгоритмі генерації.

Ключовим етапом обробки є валідація реквізитів, яка реалізована як багаторівнева система перевірки. На першому рівні виконується перевірка обов'язковості заповнення полів і відповідності їх формату заданим правилам.

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 30
Зм.	Арк.	№ докум.	Підпис	Дата		

На цьому рівні виявляються такі помилки, як відсутність необхідних значень, неправильний формат дат, некоректні числові значення або недопустимі символи. На другому рівні здійснюється перевірка логічної узгодженості між реквізитами. Наприклад, контролюється, щоб дата завершення договору не передувала даті його початку, суми були коректно обчислені, а взаємопов'язані поля не суперечили одне одному.

Додатково може виконуватися контекстна перевірка, яка враховує специфіку конкретного типу договору. У цьому випадку алгоритм перевірки адаптується до обраного шаблону та враховує особливості його структури. Це дозволяє підвищити точність виявлення помилок і забезпечити відповідність документа вимогам конкретного сценарію використання. Усі результати перевірки фіксуються, і в разі виявлення помилок система формує детальні повідомлення, що дозволяє швидко внести необхідні виправлення.

Процес генерації організовано як послідовне формування документа, у межах якого кожен блок тексту обробляється окремо. Спочатку формується внутрішнє представлення документа у вигляді структурованого тексту, після чого виконується його перетворення у фінальний формат. Це дає змогу контролювати правильність підстановки значень на кожному етапі та за потреби виконувати додаткову обробку, наприклад форматування або перевірку цілісності тексту.

Після формування тексту виконується генерація файлу у вибраному форматі, наприклад DOCX або PDF. На цьому етапі відбувається застосування стилів, форматування тексту, розміщення елементів документа та підготовка його до подальшого використання. Генерація файлу виконується таким чином, щоб забезпечити відповідність документа встановленим вимогам і можливість його відкриття у стандартних програмних засобах.

Завершальним етапом є збереження документа та супровідної інформації. Система фіксує факт створення договору, зберігає його у відповідному сховищі та записує службові дані, такі як дата створення, використаний шаблон і

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

пов'язані реквізити. Це дозволяє організувати подальший доступ до документа, відстеження історії змін і повторне використання даних. Блок-схема алгоритму генерації та перевірки договору зображено на рисунку 2.2.

У межах організації обробки даних також реалізовано механізм обробки помилок і виключних ситуацій. У разі виникнення помилок на будь-якому етапі система забезпечує їх фіксацію та формування повідомлень, що дозволяє користувачу оперативно реагувати та вносити необхідні зміни. Зазначений алгоритм підвищує стабільність роботи системи та забезпечує передбачуваність її поведінки.

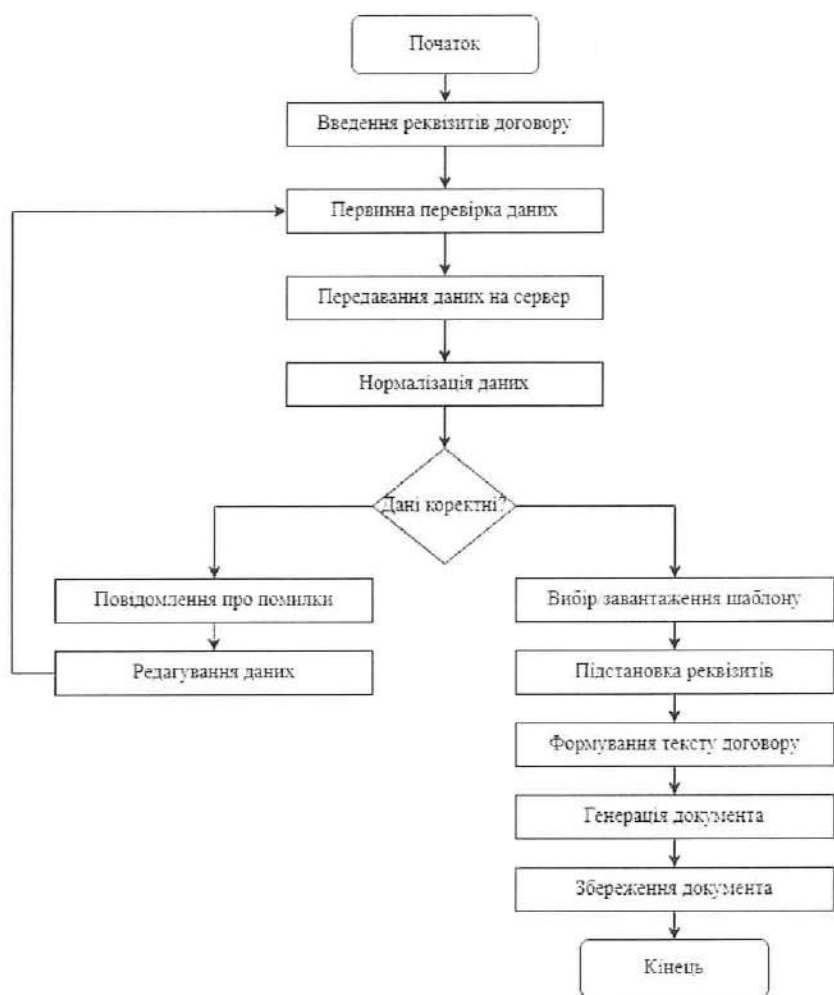


Рисунок 2.2 – Блок-схема алгоритму генерації та перевірки договору

Алгоритм роботи системи можна описати як послідовність взаємопов'язаних дій: введення даних, первинна перевірка, передавання на сервер, нормалізація, багаторівнева валідація, обробка шаблону, формування тексту, генерація файлу та його збереження. Кожен етап є необхідним і виконується у визначеному порядку, що забезпечує цілісність процесу.

Відповідно, організація обробки даних і алгоритм генерації та перевірки договорів реалізовані як цілісна система, у якій кожен компонент виконує свою функцію, а їх взаємодія забезпечує автоматизоване створення коректних і узгоджених документів. Запропонований підхід дозволяє мінімізувати вплив людського фактора, підвищити якість підготовлених договорів і створити основу для подальшого розвитку інформаційної системи.

2.5 Організація взаємодії клієнтської та серверної частин

Організація взаємодії клієнтської та серверної частин інформаційної системи автоматизованої генерації та перевірки договорів побудована на основі клієнт-серверної архітектури з використанням принципів розподілених систем. У межах цього підходу реалізовано чітке розмежування функцій між рівнем представлення даних і рівнем бізнес-логіки, що дозволяє забезпечити незалежність компонентів, підвищити масштабованість і спростити супровід системи.

Клієнтська частина реалізує рівень представлення та відповідає за взаємодію з користувачем. Вона функціонує у вигляді веб-додатку, що виконується у браузері, та забезпечує формування інтерфейсу для введення реквізитів договору, вибору шаблонів і ініціації процесу генерації документа. На цьому рівні реалізовано механізми локальної валідації даних, які дозволяють виконувати базову перевірку ще до передавання інформації на сервер. До таких перевірок належить контроль обов'язковості полів, відповідність форматів

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

введених значень і первинна обробка даних, що дозволяє зменшити кількість некоректних запитів.

Передавання даних від клієнтської частини до серверної здійснюється за допомогою HTTP-протоколу з використанням REST-підходу. Усі запити формуються у вигляді структурованих повідомлень, що передаються у форматі JSON. Такий формат дозволяє забезпечити уніфіковане представлення даних, зручність серіалізації та десеріалізації, а також спрощує інтеграцію з іншими системами. Кожен запит містить повний набір реквізитів, необхідних для виконання відповідної операції, що дозволяє серверу обробляти його незалежно від попередніх викликів.

Серверна частина функціонує як окремий програмний модуль, що реалізує набір API-методів, кожен із яких відповідає за виконання конкретної операції. Архітектурно сервер організовано за принципом модульності, де кожен запит проходить через визначений ланцюг обробки: приймання, валідація, обробка бізнес-логіки, формування відповіді. На етапі приймання виконується десеріалізація даних і перевірка їх відповідності структурі запиту. Далі виконується передавання даних до відповідних модулів системи.

Обробка запитів на сервері реалізована з використанням асинхронної моделі виконання, що дозволяє обробляти декілька запитів одночасно без блокування основного потоку виконання. Цей метод є особливо ефективним у багатокористувацькому середовищі, де одночасно можуть виконуватися запити на генерацію документів або перевірку даних. Асинхронна обробка дозволяє зменшити затримки відповіді та підвищити пропускну здатність системи.

Взаємодія між клієнтом і сервером реалізована у вигляді чітко визначеного набору кінцевих точок API. Зокрема, передбачено окремі запити для отримання списку доступних шаблонів, передавання реквізитів договору, виконання перевірки даних, ініціації генерації документа та отримання результату. Кожна операція має власний маршрут і визначений формат вхідних і вихідних даних,

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

що дозволяє забезпечити однозначність взаємодії та спростити тестування системи.

Відповіді сервера формуються у структурованому вигляді та містять або результат виконання операції, або інформацію про помилки. У разі успішного виконання запиту клієнт отримує дані про сформований документ, включаючи посилання на файл або його вміст. У разі виникнення помилки формується відповідне повідомлення, яке містить опис проблеми та вказівки щодо її усунення. Такий механізм забезпечує прозорість взаємодії та дозволяє користувачу швидко реагувати на помилки.

Особливу увагу приділено обробці помилок на всіх рівнях взаємодії. Помилки можуть виникати як на клієнтській стороні, так і на сервері, тому система передбачає централізований механізм їх обробки. На сервері всі виняткові ситуації перехоплюються та перетворюються у стандартизовані повідомлення, що передаються клієнту. Це дозволяє уникнути неконтрольованих збоїв і забезпечити стабільність роботи системи.

Для забезпечення безпеки взаємодії використовується протокол HTTPS, що гарантує шифрування переданих даних. Додатково реалізовано механізми автентифікації користувачів, які дозволяють обмежити доступ до функціоналу системи. Кожен запит до серверу може супроводжуватися токеном авторизації, що перевіряється перед виконанням операції. Діаграма взаємодії клієнта та сервера подано на рисунку 2.3. Це дозволяє забезпечити контроль доступу та захистити систему від несанкціонованого використання.

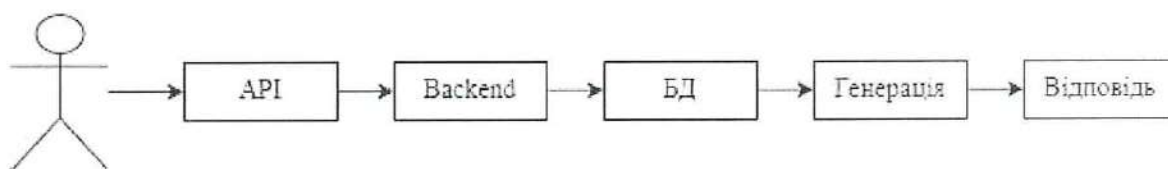


Рисунок 2.3 – Діаграма взаємодії клієнта та сервера

У межах організації взаємодії також враховано можливість кешування окремих запитів, що дозволяє зменшити навантаження на сервер і прискорити доступ до часто використовуваних даних, таких як список шаблонів. Крім того, передбачено механізми логування запитів і відповідей, що дозволяє здійснювати моніторинг роботи системи та аналізувати її поведінку в різних умовах експлуатації.

Архітектурне рішення забезпечує можливість масштабування системи за рахунок розділення клієнтської та серверної частин. У разі збільшення навантаження серверна частина може бути розгорнута на кількох вузлах, що дозволяє підвищити продуктивність і забезпечити стабільну роботу системи. Клієнтська частина, у свою чергу, може бути адаптована до різних платформ без зміни серверної логіки.

Організація взаємодії клієнтської та серверної частин реалізована як багаторівнева система обміну даними, що забезпечує ефективну, надійну та безпечну роботу інформаційної системи. Використання REST-архітектури, асинхронної обробки, структурованих форматів даних і механізмів контролю доступу створює основу для стабільного функціонування системи та її подальшого розвитку.

2.6 Особливості забезпечення коректності даних та безпеки системи

У процесі розроблення інформаційної системи автоматизованої генерації та перевірки договорів особливу увагу приділено забезпеченню коректності даних та захисту інформації, оскільки саме ці аспекти безпосередньо впливають на надійність функціонування системи та достовірність сформованих документів. Договір як юридично значущий документ повинен містити точні, узгоджені та перевірені дані, тому механізми контролю коректності реалізовано на всіх етапах обробки інформації.

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

Забезпечення коректності даних реалізовано як багаторівневу систему перевірки, що охоплює як клієнтську, так і серверну частини. На рівні інтерфейсу користувача виконано первинну валідацію введених реквізитів, яка дозволяє одразу виявляти очевидні помилки, такі як відсутність обов'язкових значень, неправильний формат даних або використання недопустимих символів. Це зменшує кількість некоректних запитів до серверної частини та підвищує зручність роботи з системою.

Основна перевірка даних виконується на серверному рівні, де реалізовано централізований механізм валідації. У цьому модулі застосовано формалізовані моделі даних, які визначають структуру реквізитів і правила їх перевірки. Кожне поле перевіряється на відповідність типу даних, діапазону допустимих значень і обов'язковості заповнення. Такий підхід уніфікує процес перевірки та дозволяє уникнути дублювання логіки в різних частинах системи.

Додатково реалізовано механізм нормалізації даних, який забезпечує приведення введеної інформації до єдиного формату. Це стосується форматів дат, числових значень, текстових полів і ідентифікаторів. Нормалізація дозволяє усунути неоднозначності та підвищити узгодженість даних, що є важливим для подальшої обробки та генерації документів.

У системі також передбачено обробку виключних ситуацій, які можуть виникати під час обробки даних. Усі помилки фіксуються та перетворюються у зрозумілі повідомлення, що передаються користувачу. Це дозволяє оперативно реагувати на проблеми та забезпечує стабільність роботи системи навіть у разі виникнення некоректних вхідних даних.

Питання безпеки системи розглядається як невід'ємна складова її архітектури. Для забезпечення захисту інформації використано комплексний підхід, що включає механізми автентифікації, авторизації та шифрування даних. Доступ до функціоналу системи обмежується лише авторизованими користувачами, що дозволяє запобігти несанкціонованому використанню.

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

Передавання даних між клієнтською та серверною частинами здійснюється з використанням захищеного протоколу, що забезпечує шифрування інформації під час обміну. Це дозволяє запобігти перехопленню або зміні даних під час передавання. Крім того, реалізовано механізми перевірки цілісності запитів, що дозволяє виявляти спроби несанкціонованого втручання.

Особливу увагу приділено захисту від типових загроз, пов'язаних із веб-застосунками. У системі реалізовано перевірку вхідних даних для запобігання ін'єкційним атакам, обмеження доступу до внутрішніх ресурсів і контроль виконання запитів. Також передбачено механізми обмеження кількості запитів, що дозволяє захистити систему від перевантаження або зловмисних дій.

З метою підвищення надійності роботи системи реалізовано логування основних подій, що відбуваються під час обробки даних. Це дозволяє відстежувати дії користувачів, аналізувати помилки та забезпечувати контроль за функціонуванням системи. Наявність журналів подій створює можливість для подальшого аналізу та вдосконалення системи.

У межах забезпечення безпеки також враховано питання зберігання даних. Конфіденційна інформація, що міститься у договорах, зберігається з урахуванням вимог до захисту, включаючи обмеження доступу та контроль змін. Це дозволяє забезпечити цілісність і захищеність інформації протягом усього життєвого циклу документа.

Як наслідок, реалізовані механізми забезпечення коректності даних та безпеки системи формують цілісний підхід до контролю якості інформації та захисту даних. Поєднання багаторівневої валідації, нормалізації, контролю доступу та захисту передавання даних дозволяє створити надійну інформаційну систему, здатну ефективно функціонувати в умовах реального використання та забезпечувати високий рівень довіри до сформованих документів.

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОМАТИЗОВАНОЇ ГЕНЕРАЦІЇ ТА ПЕРЕВІРКИ ДОГОВОРІВ

3.1 Реалізація серверної частини системи

Серверну частину інформаційної системи автоматизованої генерації та перевірки типових договорів реалізовано як основний програмний рівень, що забезпечує приймання запитів від клієнтського інтерфейсу, перевірку введених реквізитів, взаємодію з шаблонами, роботу з базою даних і передавання даних до модуля генерації документів. Саме серверна частина виконує головну логіку системи, оскільки клієнтський інтерфейс використовується переважно для введення даних, вибору шаблону та відображення результату.

Для реалізації серверного рівня використано підхід побудови вебсервісу на основі API. Така структура дозволяє відокремити користувацький інтерфейс від внутрішньої логіки обробки даних. Таке рішення дозволило клієнтській частині не звертатися безпосередньо до бази даних або файлового сховища, а всі операції виконуються через серверні маршрути. Це підвищує контрольованість роботи системи, спрощує обробку помилок і створює основу для подальшого розширення функціональності.

Серверний застосунок організовано як окремий модуль, який приймає структуровані дані у форматі JSON. Після надходження запиту система виконує перевірку отриманих реквізитів, приводить їх до потрібного формату, обирає відповідний шаблон договору та передає підготовлені дані до модуля генерації документа. У разі виявлення помилок сервер не запускає формування договору, а повертає клієнтській частині повідомлення з описом проблеми.

Початкову конфігурацію серверного застосунку реалізовано через створення основного об'єкта API, який надалі використовується для підключення маршрутів і функціональних модулів системи.

```
from fastapi import FastAPI
app = FastAPI(title="Contract Generator API")
```

Наведений фрагмент показує базову точку входу до серверної частини. У повній реалізації до цього об'єкта підключено маршрути для роботи з шаблонами, реквізитами, перевіркою даних і генерацією договорів. Цей метод дозволяє зберегти зрозумілу структуру проєкту та не перевантажувати один файл усією логікою системи.

Структуру серверної частини поділено на кілька логічних блоків. Окремо виділено модуль маршрутів API, модуль моделей даних, сервіс перевірки реквізитів, сервіс роботи з шаблонами, модуль генерації документів і модуль доступу до бази даних. Такий поділ дозволяє кожному компоненту виконувати власне завдання. Наприклад, модуль маршрутів лише приймає запити та передає їх далі, а перевірка реквізитів і генерація документа виконуються в окремих сервісах.

Для опису структури вхідних даних використано моделі, які визначають очікувані поля договору. Це дозволяє серверу працювати не з довільним набором значень, а з формалізованим об'єктом, у якому кожне поле має визначений тип. Така організація особливо важлива для системи генерації договорів, оскільки помилка в одному реквізиті може вплинути на весь зміст сформованого документа.

```
class ContractData(BaseModel):  
    client_name: str  
    contract_number: str  
    contract_date: str  
    total_amount: float
```

Цей фрагмент відображає спрощений приклад моделі реквізитів. У повній реалізації кількість полів є більшою, оскільки договір може містити інформацію про сторони, адреси, ідентифікаційні коди, банківські реквізити, строки виконання зобов'язань, суму договору та додаткові умови. Використання такої моделі дозволяє централізовано контролювати структуру даних і не допускати передавання некоректних значень у модуль генерації.

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 40
Зм.	Арк.	№ докум.	Підпис	Дата		

Одним із головних елементів серверної частини став маршрут генерації договору. Він приймає реквізити від клієнтського інтерфейсу, запускає перевірку даних, обирає шаблон і повертає результат виконання операції. У спрощеному вигляді логіка такого маршруту подана нижче.

```
@app.post("/generate")
def generate_contract(data: ContractData):
    file_path = create_document(data)
    return {"status": "success", "file": file_path}
```

У цьому прикладі показано загальний принцип роботи маршруту. Після отримання реквізитів викликається функція створення документа, а клієнтська частина отримує відповідь із результатом. У реальній роботі перед створенням файлу додатково виконується валідація реквізитів, обробка можливих виняткових ситуацій і запис службової інформації про сформований договір.

Важливою частиною серверної реалізації є обробка помилок. Якщо користувач не заповнив обов'язкове поле, ввів некоректну дату або передав значення у неправильному форматі, сервер формує відповідь із повідомленням про помилку. Таким чином, система не переходить до етапу генерації документа з неповними або неправильними даними. Такий механізм підвищує надійність роботи та зменшує кількість помилкових договорів.

Серверна частина також забезпечує доступ до шаблонів договорів. Під час запиту на генерацію система визначає, який саме шаблон потрібно використати, отримує його зі сховища або бази даних, після чого передає разом із реквізитами до модуля формування документа. Це дозволяє централізовано керувати шаблонами та не дублювати їх у різних частинах системи.

Окрему роль виконує взаємодія з базою даних. У базі зберігається інформація про шаблони, створені договори, дату генерації, користувача та службові параметри. Серверний рівень відповідає за виконання запитів до бази, отримання потрібних даних і збереження результатів роботи. Як наслідок, система може не лише сформувати договір, а й зберегти історію його створення для подальшого перегляду або повторного використання.

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 41
Зм.	Арк.	№ докум.	Підпис	Дата		

На рисунку 3.1 розміщено структуру серверної частини системи, де показано взаємодію API, модуля перевірки реквізитів, модуля шаблонів, модуля генерації документів і бази даних.



Рисунок 3.1 – Структура серверної частини інформаційної системи генерації договорів

Реалізована серверна частина також передбачає можливість подальшого розширення. За потреби до системи можна додати нові маршрути, нові типи договорів, додаткові правила перевірки або інші формати експорту документів. Модульна побудова дозволяє вносити такі зміни без повного переписування серверної логіки.

Серверну частину системи реалізовано як центральний програмний компонент, що поєднує клієнтський інтерфейс, базу даних, шаблони договорів, механізм перевірки реквізитів і модуль генерації документів. Така реалізація забезпечує стабільну обробку запитів, контроль коректності даних і формування готового договору у визначеному форматі.

3.2 Реалізація структури бази даних та зберігання інформації

Реалізацію структури бази даних інформаційної системи автоматизованої генерації та перевірки типових договорів виконано з урахуванням того, що система працює не лише з готовими файлами договорів, а й з усіма даними, які беруть участь у процесі їх створення. До таких даних належать шаблони документів, реквізити сторін, відомості про створені договори, службова інформація про дату формування документа, формат збереження, статус обробки та користувача, який ініціює створення договору. Через це базу даних розглянуто не просто як місце збереження інформації, а як важливу складову всієї логіки системи, що забезпечує зв'язок між шаблонами, введеними реквізитами та сформованими документами.

Для організації зберігання інформації застосовано реляційну модель даних, оскільки вона добре підходить для роботи зі структурованими сутностями та зв'язками між ними. У межах розробленої системи кожен договір пов'язано з певним шаблоном, набором реквізитів, користувачем і результатом генерації. Така логіка природно відображається у вигляді таблиць, між якими встановлено зв'язки за допомогою первинних і зовнішніх ключів. Це дозволило уникнути дублювання даних, забезпечити цілісність інформації та сформувати зрозумілу структуру для подальшої обробки договорів.

Основними сутностями бази даних визначено таблиці користувачів, шаблонів, договорів, реквізитів і сформованих документів. Таблицю користувачів використано для зберігання облікової інформації та ідентифікації особи, яка створює або редагує договір. Таблицю шаблонів застосовано для збереження відомостей про тип договору, назву шаблону, шлях до файлу шаблону або його текстове представлення. Центральною сутністю виступає таблиця договорів, оскільки саме вона поєднує користувача, обраний шаблон, введені реквізити та результат генерації.

Окремо реалізовано таблицю реквізитів, у якій збережено дані, що підставляються у відповідні поля шаблону договору. До таких даних віднесено назву сторони, ідентифікаційний код, адресу, банківські реквізити, номер договору, дату укладання, суму договору та інші параметри, які залежать від конкретного типу документа. Винесення реквізитів в окрему структуру дозволило не перевантажувати основну таблицю договорів і зробити модель даних більш гнучкою. Це особливо важливо для системи, у якій надалі може бути додано нові типи договорів із різним набором полів.

Для зберігання інформації про сформовані договори створено окрему таблицю документів. У ній зафіксовано назву згенерованого файлу, його формат, шлях до місця збереження, дату створення та зв'язок із відповідним договором. Запропоноване рішення дозволить відокремити логічний запис про договір від фізичного файлу, який зберігається у файловому сховищі. У результаті система може працювати з кількома форматами одного документа, наприклад DOCX і PDF, не порушуючи загальної структури бази даних.

Спрощений приклад структури таблиці договорів подано у фрагменті коду:

```
CREATE TABLE contracts (  
    id INTEGER PRIMARY KEY,  
    user_id INTEGER,  
    template_id INTEGER,  
    contract_number TEXT,  
    created_at DATETIME,  
    status TEXT  
);
```

У наведеному фрагменті показано загальний принцип організації таблиці договорів. У ній передбачено унікальний ідентифікатор запису, посилання на користувача, посилання на шаблон, номер договору, дату створення та поточний статус документа. У повній реалізації таблицю може бути доповнено додатковими полями, зокрема датою останнього редагування, типом договору, коментарем або службовим ідентифікатором сформованого файлу.

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 44
Зм.	Арк.	№ докум.	Підпис	Дата		

Зв'язки між таблицями реалізовано так, щоб кожна сутність мала чітке місце у загальній структурі системи. Один користувач може створити багато договорів, один шаблон може використовуватися для формування багатьох договорів, а кожен договір може містити набір реквізитів і один або кілька згенерованих документів. Така організація відповідає логіці роботи системи та дозволяє зручно виконувати пошук, фільтрацію, перегляд історії створення документів і повторне використання наявних даних.

На рисунку 3.2 подано структуру таблиць бази даних інформаційної системи та зв'язки між ними. У схемі відображено таблиці users, templates, contracts, requisites та documents, а також показано їх взаємодію через зовнішні ключі. Це дозволяє наочно пояснити, як саме організовано зберігання інформації в системі та яким чином окремі сутності пов'язані між собою.

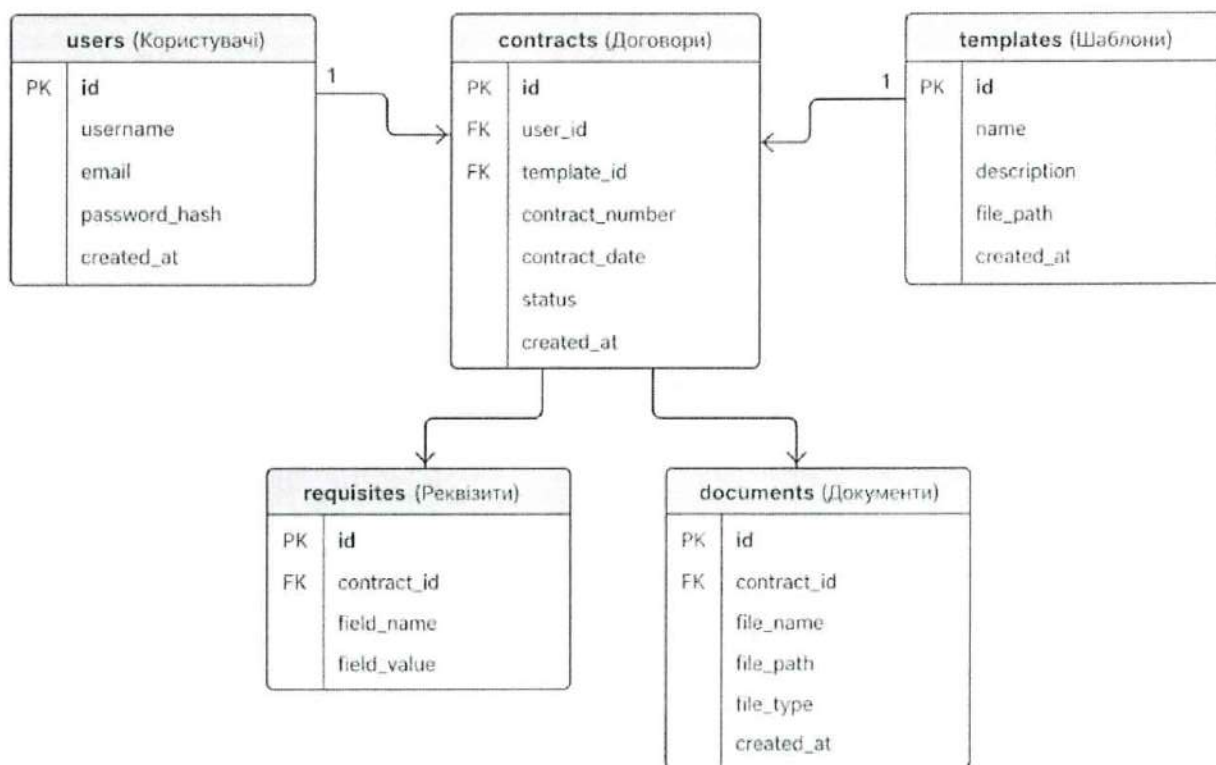


Рисунок 3.2 – Структура бази даних інформаційної системи генерації договорів

Окрім зберігання структурованих записів, у системі реалізовано роботу з файловим сховищем. Сам вміст сформованого DOCX або PDF-файлу не збережено безпосередньо в таблицях бази даних, оскільки це могло б ускладнити її структуру та збільшити обсяг збережених даних. Натомість у таблиці документів збережено шлях до файлу, його формат і службові відомості. Фізичний файл розміщено у відповідному каталозі файлового сховища. Така реалізація спростить доступ до готових документів і водночас зберігає базу даних достатньо легкою для обробки.

Для підвищення зручності зберігання сформовані документи розміщено у каталогах за датою створення або типом договору. Наприклад, окремо можуть зберігатися договори надання послуг, договори оренди або договори купівлі-продажу. Це дозволяє підтримувати порядок у файловому сховищі навіть у разі збільшення кількості створених документів. При цьому база даних залишається основним джерелом службової інформації про те, який файл належить до конкретного договору.

Важливим аспектом реалізації бази даних стало забезпечення цілісності інформації. Для цього між таблицями передбачено зв'язки, які не дозволяють створити договір без відповідного шаблону або зберегти документ без зв'язку з конкретним договором. Це запобігає появі зайвих або непов'язаних записів і забезпечує логічну узгодженість структури даних. Якщо договір сформовано на основі певного шаблону, цю інформацію збережено в базі та надалі її можна використати для перегляду, повторної генерації або аналізу.

Окремо реалізовано зберігання статусу договору. Статус дозволяє відстежувати, на якому етапі перебуває документ: створений, перевірений, згенерований або збережений. Така службова інформація є корисною для подальшого розвитку системи, оскільки на її основі можна реалізувати механізми погодження, повторного редагування або контролю версій. Навіть у базовій реалізації статус допомагає краще відображати результат роботи серверної частини та стан документа в інтерфейсі користувача.

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

Під час реалізації зберігання реквізитів враховано потребу в подальшому розширенні структури даних. Різні типи договорів можуть мати різний набір полів, тому модель реквізитів зроблено достатньо гнучкою. У системі передбачено можливість зберігати реквізити як окремі поля або як структурований набір значень, що пов'язаний із конкретним договором. Це дозволяє додавати нові шаблони без повної перебудови бази даних і зберігати логіку роботи системи стабільною.

У межах роботи з базою даних реалізовано базові операції створення, читання, оновлення та видалення записів. Серверна частина звертається до бази під час отримання списку шаблонів, створення нового договору, запису реквізитів, збереження інформації про сформований документ і подальшого перегляду результатів. Завдяки цьому забезпечено повний цикл роботи з інформацією, необхідною для автоматизованої генерації та перевірки договорів.

Структуру бази даних реалізовано так, щоб забезпечити надійне зберігання шаблонів, реквізитів, договорів і сформованих документів. Поєднання реляційної моделі з файловим сховищем дозволило зберігати як службову інформацію, так і готові файли договорів у зручному для подальшого використання вигляді. Така організація даних створила основу для стабільної роботи системи, спростила супровід програмного рішення та забезпечила можливість його подальшого розширення.

3.3 Реалізація механізму перевірки введених реквізитів

Механізм перевірки введених реквізитів реалізовано як один із ключових компонентів інформаційної системи автоматизованої генерації та перевірки типових договорів. Його основне призначення полягає у тому, щоб не допустити формування договору з неповними, некоректними або логічно неузгодженими даними. Оскільки реквізити сторін, дати, суми, адреси та інші параметри безпосередньо підставляються у текст документа, помилка на цьому етапі може

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 47
Зм.	Арк.	№ докум.	Підпис	Дата		

призвести до неправильного змісту всього договору. Через це перевірку даних реалізовано не як допоміжну функцію, а як обов'язковий проміжний етап між введенням інформації користувачем і генерацією готового документа.

У системі передбачено багаторівневу перевірку реквізитів. Перший рівень реалізовано на стороні клієнтського інтерфейсу, де виконано базовий контроль введення даних. На цьому етапі перевірено наявність обов'язкових полів, правильність заповнення простих значень і відповідність окремих реквізитів очікуваному формату. Наприклад, система не дозволяє передати на сервер порожні значення у полях номера договору, дати укладання, назви сторони або суми договору. Це дозволяє зменшити кількість помилкових запитів і зробити взаємодію користувача із системою більш зрозумілою.

Основну перевірку реквізитів реалізовано на серверному рівні, оскільки саме серверна частина відповідає за достовірність даних перед їх використанням у шаблоні договору. Цей метод є більш надійним, оскільки клієнтська перевірка може бути обмеженою або обійденою, а серверна логіка залишається контрольованою частиною системи. Після отримання запиту сервер перевіряє всі реквізити відповідно до визначених правил, приводить їх до єдиного формату та лише після цього передає до модуля генерації документа.

Для опису структури реквізитів використано формалізовані моделі даних. Кожне поле має визначений тип, наприклад текстовий, числовий або датований. Це дозволяє автоматично відхиляти дані, які не відповідають очікуваному типу. Наприклад, у поле суми договору не може бути передано довільний текст, а в поле дати не може бути передано значення, яке не розпізнається як дата. У результаті система отримує не хаотичний набір введених користувачем символів, а структурований об'єкт, придатний для подальшої обробки.

Спрощений фрагмент перевірки реквізитів подано нижче:

```
def validate_contract(data):  
    if not data.client_name:  
        return "Не вказано назву сторони"  
    if data.total_amount <= 0:
```

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 48
Зм.	Арк.	№ докум.	Підпис	Дата		

```
return "Сума договору має бути більшою за нуль"  
return "Дані коректні"
```

Наведений фрагмент демонструє загальний принцип роботи перевірки. У повній реалізації кількість правил є більшою, оскільки система має контролювати не лише наявність значень, а й їх формат, допустимість, логічні зв'язки та відповідність обраному шаблону договору. Такий підхід дозволяє поступово розширювати набір перевірок без зміни загальної архітектури системи.

Особливий акцент зроблено на перевірці обов'язкових реквізитів. Для кожного типу договору визначено перелік полів, без яких документ не може бути сформований. До таких полів належать номер договору, дата укладання, найменування сторін, ідентифікаційні дані, адреси, предмет договору та фінансові показники. Якщо хоча б одне з обов'язкових полів не заповнено, система зупиняє процес генерації та формує повідомлення про помилку. Це дозволяє уникнути ситуацій, коли в готовому документі залишаються порожні місця або службові маркери шаблону.

Крім перевірки наявності значень, реалізовано контроль формату реквізитів. Для дат передбачено приведення до єдиного формату, що забезпечує однакове подання інформації у всіх сформованих договорах. Для числових значень виконано перевірку на коректність введення та недопущення від'ємних або нульових сум там, де це суперечить змісту документа. Для текстових полів виконано очищення від зайвих пробілів і недопустимих символів, що покращує якість підстановки даних у шаблон.

Важливим елементом механізму перевірки стала логічна валідація реквізитів. На відміну від простої перевірки формату, вона дозволяє виявляти помилки, пов'язані зі змістом документа. Наприклад, система контролює, щоб дата завершення дії договору не була ранішою за дату його укладання, а загальна сума договору відповідала заданим умовам. Якщо певні поля пов'язані між собою, їх перевірено не окремо, а в межах спільного контексту. Це дозволяє

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 49
Зм.	Арк.	№ докум.	Підпис	Дата		

підвищити якість сформованого документа та зменшити ризик внутрішніх суперечностей.

Також реалізовано перевірку відповідності реквізитів обраному шаблону. Різні типи договорів можуть містити різні набори обов'язкових полів, тому система не використовує однакові правила для всіх документів. Після вибору шаблону визначається перелік реквізитів, які мають бути заповнені саме для цього типу договору. Наприклад, для договору надання послуг важливими є строки виконання робіт і вартість послуг, а для договору оренди додатково можуть перевірятися дані про об'єкт оренди та строк користування. Це дозволяє зробити механізм перевірки більш гнучким і придатним для подальшого розширення.

У межах реалізації механізму перевірки передбачено нормалізацію даних. Перед збереженням і передаванням у шаблон система приводить реквізити до єдиного вигляду. Зайві пробіли видаляються, дати перетворюються до узгодженого формату, числові значення приводяться до стандартного запису, а текстові поля очищуються від випадкових символів. Відповідно, зменшується ймовірність появи різного написання однакових даних у різних частинах договору.

Повідомлення про помилки сформовано так, щоб користувач міг зрозуміти, яке саме поле потребує виправлення. Замість загального повідомлення про некоректність даних система повертає конкретний опис проблеми, наприклад про незаповнене поле, неправильний формат дати або недопустиме значення суми. Така організація зворотного зв'язку дозволяє швидко виправити помилки та повторно запустити процес перевірки без зайвих дій.

На рисунку 3.3 подано схему роботи механізму перевірки введених реквізитів. У схемі відображено послідовність етапів від отримання даних із клієнтського інтерфейсу до прийняття рішення про можливість генерації договору. Окремо показано гілку обробки помилок, за якої система повертає

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

користувачу повідомлення та не допускає формування документа з некоректними даними.



Рисунок 3.3 – Схема перевірки введених реквізитів договору

Після успішного проходження всіх етапів перевірки реквізити передаються до модуля генерації договорів. Це означає, що в шаблон потрапляють лише ті дані, які відповідають встановленим правилам і не містять критичних помилок. Це забезпечує зв'язок між механізмом перевірки та модулем формування документа, а також підвищує загальну надійність системи.

Отже, механізм перевірки введених реквізитів реалізовано як багаторівневу систему контролю, що охоплює перевірку обов'язкових полів,

Зм.	Арк.	№ докум.	Підпис	Дата

форматів, логічних зв'язків, відповідності шаблону та нормалізацію даних. Це дозволило зменшити ймовірність помилок у сформованих договорах, підвищити якість підготовки документів і забезпечити стабільну роботу системи перед етапом генерації готового файлу.

3.4 Реалізація генерації договорів на основі шаблонів

Генерацію договорів на основі шаблонів реалізовано як один із центральних функціональних процесів інформаційної системи. Саме цей механізм забезпечує перетворення введених і перевічених реквізитів у готовий документ, який має узгоджену структуру, заповнені поля та придатний формат для подальшого використання. На відміну від ручного редагування текстового файлу, у розробленій системі формування договору виконано автоматизовано, тобто користувач не змінює текст документа безпосередньо, а лише вводить необхідні дані у відповідні поля інтерфейсу.

Основою генерації договорів виступає шаблон документа. У системі шаблон розглянуто як заздалегідь підготовлений файл або текстова структура, у якій сталі фрагменти договору поєднано зі спеціальними маркерами для підстановки реквізитів. До сталих фрагментів належать назва договору, типові формулювання розділів, загальні положення, порядок виконання зобов'язань, відповідальність сторін і заключні положення. Змінними елементами виступають номер договору, дата укладання, найменування сторін, адреси, сума договору, строк дії, банківські реквізити та інші параметри, які залежать від конкретної ситуації.

Перед запуском генерації система вже має отримати реквізити, які пройшли перевірку на коректність. Це важливий момент, оскільки модуль генерації не повинен працювати з неповними або помилковими даними. У загальній логіці системи генерація договору розташована після етапів введення, нормалізації та валідації реквізитів. За рахунок цього в шаблон передаються

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

лише підготовлені значення, що зменшує ризик появи порожніх полів, неправильних дат або неузгоджених сум у фінальному документі.

Механізм генерації реалізовано за принципом підстановки значень у шаблон. Для цього у шаблоні передбачено спеціальні позначення, які відповідають полям моделі даних. Наприклад, маркер номера договору замінюється значенням, введеним у відповідному полі, маркер назви сторони – найменуванням контрагента, а маркер суми – числовим значенням вартості договору. Завдяки такому підходу один і той самий шаблон можна використовувати багаторазово, формуючи на його основі різні договори.

Спрощений приклад підстановки реквізитів у шаблон подано нижче:

```
context = {  
  "contract_number": data.contract_number,  
  "client_name": data.client_name,  
  "total_amount": data.total_amount  
}
```

У наведеному фрагменті показано принцип формування словника даних, який передається до шаблону. Кожен ключ відповідає певному маркеру в документі, а значення береться з перевірених реквізитів. У повній реалізації такий набір є ширшим і містить усі дані, необхідні для формування конкретного типу договору.

Для роботи з шаблонами застосовано підхід, за якого структура документа відокремлена від програмної логіки. Це означає, що текст договору не зберігається безпосередньо в коді програми, а винесений в окремі шаблонні файли. Така організація значно спрощує супровід системи, оскільки зміна формулювань у договорі не потребує повного переписування програмного коду. Достатньо змінити відповідний шаблон, після чого система зможе використовувати його для подальшої генерації документів.

Окремою перевагою шаблонного підходу є підтримка різних типів договорів. У системі може бути передбачено кілька шаблонів, наприклад договір надання послуг, договір оренди, договір купівлі-продажу або договір підряду. Кожен із них має власну структуру та набір необхідних реквізитів. Після вибору типу договору система завантажує відповідний шаблон і застосовує до нього

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 53
Зм.	Арк.	№ докум.	Підпис	Дата		

саме ті дані, які потрібні для цього документа. Це дозволяє зробити систему більш гнучкою та придатною для використання у різних робочих ситуаціях.

Генерацію документа реалізовано у кілька послідовних етапів. Спочатку система отримує ідентифікатор обраного шаблону та завантажує його з файлового сховища або бази даних. Далі формується набір реквізитів у вигляді структурованого об'єкта. Після цього виконується підстановка значень у шаблон і формування внутрішнього представлення документа. На завершальному етапі система створює файл у потрібному форматі та зберігає його у визначеному каталозі.

У процесі формування документа враховано не лише просту заміну маркерів, а й можливість використання умовних блоків. Це потрібно для тих випадків, коли окремі частини договору мають додаватися лише за певних умов. Наприклад, якщо в договорі передбачено авансовий платіж, система може додати відповідний пункт до тексту. Якщо така умова відсутня, цей блок не включається у фінальний документ. Завдяки цьому шаблон стає не просто статичною формою, а більш гнучким інструментом формування тексту.

Спрощений приклад виклику генерації документа можна подати так:

```
doc.render(context)
doc.save(output_path)
```

Цей фрагмент демонструє загальну ідею роботи модуля генерації: спочатку шаблон заповнюється підготовленими даними, після чого сформований документ зберігається у визначеному місці. У повній реалізації перед збереженням додатково виконується формування назви файлу, перевірка шляху збереження та запис інформації про документ до бази даних.

Назву сформованого файлу організовано так, щоб вона була зрозумілою та не створювала конфліктів у файловому сховищі. Для цього до назви можуть включатися номер договору, дата створення або унікальний ідентифікатор запису. Описаний принцип дозволяє уникнути перезапису раніше створених файлів і спрощує подальший пошук документа. Наприклад, файл може зберігатися з назвою, що містить тип договору, номер і дату формування.

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
						54
Зм.	Арк.	№ докум.	Підпис	Дата		

Після створення документа система зберігає не лише сам файл, а й службову інформацію про результат генерації. У базі даних фіксується зв'язок між договором, використаним шаблоном, реквізитами та сформованим файлом. Це дозволяє в майбутньому переглянути історію створення договору, повторно отримати файл або сформувати нову версію документа на основі тих самих даних. Така організація є важливою для забезпечення цілісності інформації та контролю результатів роботи системи.

На рисунку 3.4 подано послідовність генерації договору на основі шаблону. У схемі відображено отримання перевірених реквізитів, вибір шаблону, підстановку значень, формування документа, збереження файлу та передавання результату користувачу. Це дозволяє наочно показати, як саме реалізовано процес створення договору в межах розробленої системи.



Рисунок 3.4 – Послідовність генерації договору на основі шаблону

Окрему увагу приділено обробці ситуацій, коли шаблон не знайдено або окремі маркери не мають відповідних значень. У таких випадках система не повинна формувати неповний документ. Замість цього повертається повідомлення про помилку, а процес генерації зупиняється. Це дозволяє уникнути створення файлів із незаповненими службовими позначеннями та підвищує якість фінального результату.

У системі також враховано потребу у формуванні документів у поширених форматах. Основним форматом для редагованого договору визначено DOCX, оскільки він зручний для подальшого перегляду, редагування та використання у звичних офісних програмах. За потреби документ може бути додатково перетворено у PDF, що є зручним для надсилання, архівування або передавання на підпис. Це дає поєднати гнучкість редагованого документа з практичністю фінального незмінного файлу.

Під час реалізації генерації договорів також враховано питання збереження форматування. Для договорів важливо, щоб текст мав однаковий вигляд, містив коректні відступи, заголовки, нумерацію пунктів і реквізити сторін у відповідних місцях. Через це шаблон готується заздалегідь із потрібною структурою, а система лише підставляє значення у визначені поля. Це дозволяє зберегти єдиний стиль оформлення для всіх сформованих договорів і зменшити кількість ручних правок після генерації.

Після завершення генерації серверна частина повертає клієнтському інтерфейсу інформацію про успішне створення документа. Користувач отримує можливість завантажити сформований договір або переглянути його у списку створених документів. Якщо під час генерації виникла помилка, клієнтська частина отримує відповідне повідомлення, яке пояснює причину невдалого виконання операції. Це забезпечує зрозумілу взаємодію між користувачем і системою.

У кінцевому результаті генерацію договорів на основі шаблонів реалізовано як послідовний автоматизований процес, у якому перевірені

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

реквізити підставляються у відповідний шаблон, після чого формується готовий документ у визначеному форматі. Така реалізація дозволила зменшити обсяг ручної роботи, забезпечити однакове оформлення договорів, підвищити точність підстановки даних і створити основу для подальшого розширення набору підтримуваних шаблонів.

3.5 Реалізація клієнтського інтерфейсу користувача

Клієнтський інтерфейс інформаційної системи автоматизованої генерації та перевірки типових договорів реалізовано як веборієнтовану частину програмного рішення, через яку користувач взаємодіє з основними функціями системи. Саме інтерфейс забезпечує введення реквізитів, вибір шаблону договору, запуск перевірки даних, формування документа та отримання результату у вигляді готового файлу. На відміну від серверної частини, яка виконує основну логіку обробки, клієнтський рівень зосереджено на зручності роботи користувача та зрозумілому поданні всіх етапів підготовки договору.

Під час реалізації клієнтського інтерфейсу враховано, що система повинна бути простою для використання навіть у тих випадках, коли користувач не має спеціальної технічної підготовки. Через це основну увагу приділено логічній послідовності елементів, зрозумілому розміщенню полів і мінімізації зайвих дій. Інтерфейс побудовано так, щоб користувач послідовно проходив усі етапи роботи: спочатку обирав тип договору, потім вводив реквізити, запускав перевірку, після цього формував документ і отримував готовий файл.

Основною сторінкою системи виступає робоча форма створення договору. У ній передбачено блок вибору шаблону, поля для введення основних реквізитів і кнопки керування процесом. Поля згруповано за змістом, що дозволяє уникнути хаотичного введення даних. Наприклад, окремо розміщено відомості про договір, дані сторін, фінансові параметри та додаткові умови. Такий поділ спростив орієнтацію в інтерфейсі та зробив процес заповнення документа більш упорядкованим.

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

зробив роботу з різними видами документів більш гнучкою, оскільки кожен шаблон має власний набір реквізитів і власну логіку заповнення.

Після вибору шаблону користувач переходить до заповнення реквізитів. У цій частині інтерфейсу реалізовано поля для введення номера договору, дати укладання, найменування сторін, адрес, ідентифікаційних кодів, банківських реквізитів, суми договору та інших параметрів.

У клієнтському інтерфейсі реалізовано первинну перевірку введених даних. Вона не замінює повноцінну серверну валідацію, але дозволяє швидко виявити найочевидніші помилки ще до надсилання запиту. Це покращує зручність роботи, оскільки користувач не очікує відповіді сервера для виправлення простих помилок.

На рисунку 3.6 подано приклад форми введення реквізитів договору. У формі відображено поділ полів за логічними групами, що дозволяє послідовно заповнити відомості про сторони, предмет договору, суму та строк дії документа.

Створення договору

Шаблон договору
 договір надання послуг

Номер договору: 15/2026 Дата укладання: 03.05.2026

Початок дії: 03.05.2026 Завершення дії: 02.06.2026

Сторона А
 ТОВ «Приклад Сервіс» 12345678
 м. Хмельницький, вул. Інститутська, 1 UA123456789000000000000001

Сторона Б
 ФОП Іваненко Іван Іванович 87654321
 м. Київ, вул. Центральна, 10 UA987654321000000000000002

Умови договору
 Предмет договору: розроблення та впровадження програмного рішення для автоматизації документообігу

Сума договору, грн: 25000 Порядок оплати: оплата здійснюється протягом 5 банківських днів після підписання акта

Рисунок 3.6 – Форма введення реквізитів договору

Після заповнення форми користувач запускає перевірку та генерацію договору. На цьому етапі клієнтська частина формує структурований запит і передає його до серверної частини. Дані передаються у форматі JSON, що забезпечує зручний обмін інформацією між інтерфейсом і сервером. У разі успішної перевірки сервер повертає відповідь про сформований документ, а інтерфейс відображає результат у зручному вигляді.

Якщо під час перевірки сервер виявляє помилки, клієнтський інтерфейс відображає повідомлення з поясненням причини. Такі повідомлення не подано у вигляді загального тексту про невдалу операцію, а прив'язано до конкретних проблемних полів. Наприклад, система може повідомити про відсутність дати договору, некоректну суму або незаповнені реквізити сторони. У результаті користувач може швидко виправити помилки та повторно запустити перевірку.

Важливим елементом інтерфейсу стало відображення стану виконання операції. Під час генерації документа користувач отримує повідомлення про те, що запит обробляється. Це потрібно для уникнення повторного натискання кнопки генерації та для кращого розуміння поточного стану системи. Після завершення обробки інтерфейс відображає результат: успішно створений документ або список помилок, які потребують виправлення.

На рисунку 3.7 подано приклад відображення результату перевірки реквізитів. У ньому показано, як система повідомляє користувача про знайдені помилки та не дозволяє перейти до генерації документа до їх усунення.

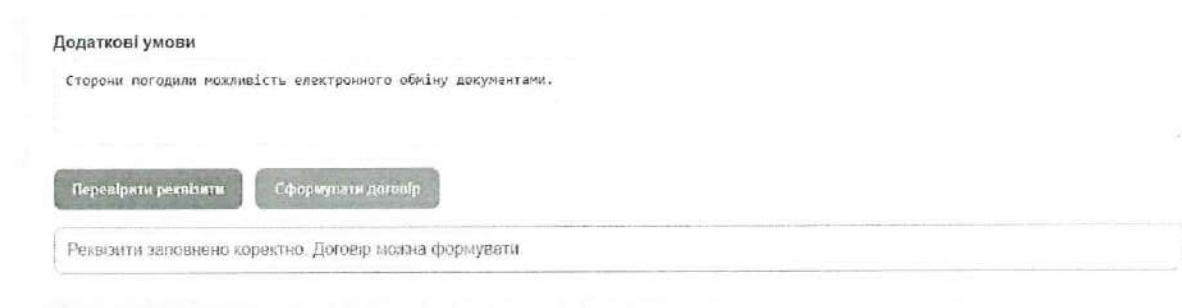


Рисунок 3.7 – Відображення результатів перевірки введених реквізитів

Після успішної генерації договору користувач отримує можливість завантажити готовий файл. У клієнтському інтерфейсі передбачено відображення посилання або кнопки для завантаження документа у форматі DOCX або PDF. Така організація дозволяє завершити процес створення договору без переходу до інших програмних засобів. Користувач вводить дані, запускає обробку та одразу отримує результат у вигляді готового документа.

Окремо реалізовано можливість перегляду списку створених документів. Такий список може містити номер договору, тип шаблону, дату створення, статус і посилання на сформований файл. Це дозволяє користувачу швидко знайти раніше створений документ і повторно завантажити його за потреби. Для подальшого розвитку системи така частина інтерфейсу може стати основою для реалізації історії змін, повторної генерації або пошуку договорів за реквізитами.

На рисунку 3.8 подано приклад сторінки з результатом генерації договору. У ній відображено повідомлення про успішне створення документа та кнопку для завантаження сформованого файлу.

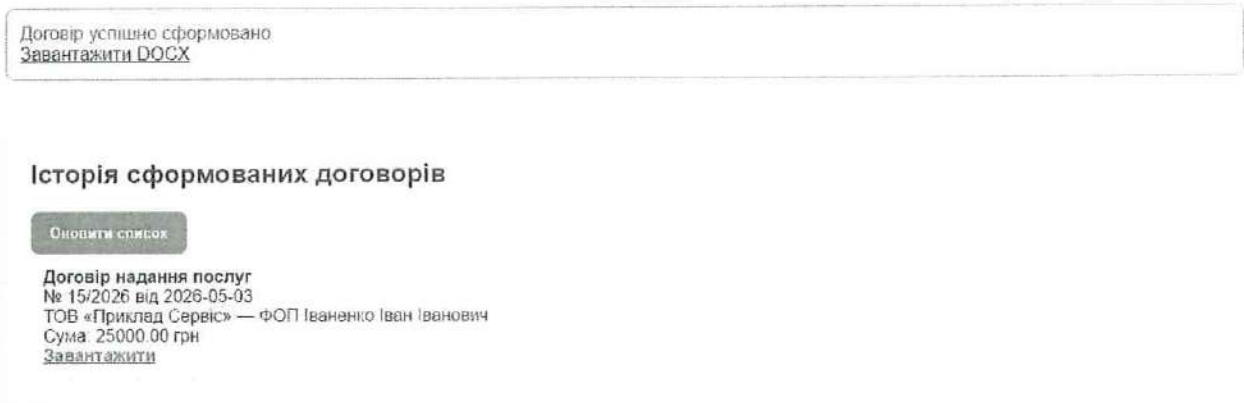


Рисунок 3.8 – Вікно результату генерації договору

З погляду технічної реалізації клієнтська частина взаємодіє з сервером через API-запити. Для отримання списку шаблонів, передавання реквізитів і завантаження результату передбачено окремі звернення до серверної частини. Така організація забезпечила чітке розмежування між інтерфейсом і логікою

					КВРІСТ.230156.23.01.05 ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

обробки даних. Якщо надалі виникне потреба змінити дизайн інтерфейсу, серверна логіка може залишатися незмінною.

У клієнтському інтерфейсі також враховано питання зручності та простоти візуального сприйняття. Надмірну кількість декоративних елементів не використано, оскільки головне завдання інтерфейсу полягає не в складному візуальному оформленні, а в швидкому та зрозумілому створенні договору. Основні кнопки, поля та повідомлення розміщено так, щоб користувач міг легко визначити наступну дію. Це особливо важливо для системи, яка працює з формами та великою кількістю реквізитів.

Для зменшення ризику випадкових помилок передбачено візуальне виділення обов'язкових полів. Якщо поле не заповнено або містить некоректне значення, інтерфейс відображає відповідне повідомлення. Такий підхід підвищує зрозумілість роботи системи та скорочує кількість повторних спроб генерації документа. У результаті користувач отримує не просто форму введення, а інструмент, який супроводжує його на всіх етапах підготовки договору.

Підсумовуючи клієнтський інтерфейс реалізовано як зручний вебрівень інформаційної системи, що забезпечує вибір шаблону, введення реквізитів, первинну перевірку даних, запуск генерації та отримання готового документа. Його структура побудована так, щоб зменшити кількість ручних дій, зробити процес підготовки договору послідовним і забезпечити зрозумілий зворотний зв'язок у разі помилок. Така реалізація створила практичну основу для ефективної взаємодії користувача з розробленою системою.

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
						62
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У кваліфікаційній роботі розглянуто проблему автоматизації процесу підготовки та перевірки типових договорів на основі введених реквізитів і заздалегідь підготовлених шаблонів. Актуальність обраної теми пов'язана з тим, що в багатьох організаціях договірна документація все ще формується вручну або з використанням напівавтоматизованих інструментів, що підвищує ризик помилок, дублювання даних і неузгодженості між окремими частинами документа. Саме тому створення інформаційної системи, яка поєднує введення реквізитів, їх перевірку та автоматизовану генерацію договору, має практичне значення для підвищення ефективності роботи з документами.

У першому розділі проаналізовано предметну область, пов'язану з підготовкою, оформленням і обробкою типових договорів. Розглянуто поняття типового договору в інформаційних системах, визначено його основні структурні елементи та показано роль шаблонів і реквізитів у процесі автоматизованого формування документа. Також проаналізовано існуючі підходи до створення та обробки договорів, зокрема корпоративні CLM-системи, комерційні сервіси для роботи з документами, офісні шаблони та програмні бібліотеки для генерації файлів. Проведений аналіз дозволив встановити, що наявні рішення або є надто складними для невеликих прикладних задач, або не забезпечують достатнього рівня перевірки реквізитів і логічної узгодженості документа.

У другому розділі розроблено загальну архітектуру інформаційної системи автоматизованої генерації та перевірки договорів. Запропоноване рішення побудовано на основі клієнт-серверної архітектури, у якій клієнтська частина відповідає за взаємодію з користувачем, а серверна частина виконує основну логіку обробки даних, перевірки реквізитів і формування документа. Обґрунтовано вибір технічних і програмних засобів реалізації, зокрема використання Python, FastAPI, SQLite, HTML, CSS, JavaScript та бібліотек для

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 63
Зм.	Арк.	№ докум.	Підпис	Дата		

роботи з DOCX-документами. Також розроблено структуру бази даних, логіку зберігання шаблонів, реквізитів і сформованих договорів, а також описано алгоритм генерації та перевірки документа.

У третьому розділі реалізовано основні компоненти інформаційної системи. Серверну частину створено як центральний програмний рівень, що приймає запити від клієнтського інтерфейсу, перевіряє введені дані, взаємодіє з базою даних і запускає процес генерації договору. Реалізовано структуру бази даних для зберігання користувачів, шаблонів, договорів, реквізитів і сформованих документів. Окремо реалізовано механізм перевірки введених реквізитів, який охоплює контроль обов'язкових полів, форматів дат, числових значень, кодів сторін і логічних зв'язків між окремими даними. Це дозволило запобігти формуванню договорів із неповною або некоректною інформацією.

Практичну частину роботи завершено реалізацією модуля генерації договорів на основі шаблонів і клієнтського вебінтерфейсу. Система забезпечує вибір типу договору, введення реквізитів, перевірку даних, формування DOCX-файлу та збереження інформації про створений документ. Клієнтський інтерфейс побудовано у вигляді зручної форми, яка дозволяє послідовно заповнити дані сторін, умови договору, фінансові показники та додаткові відомості. Після успішної перевірки система формує готовий документ і надає можливість його завантаження.

У результаті виконання кваліфікаційної роботи створено працездатний програмний прототип інформаційної системи, який демонструє можливість автоматизованої генерації та перевірки типових договорів. Розроблене рішення дозволяє зменшити обсяг ручної роботи, підвищити якість заповнення реквізитів, уніфікувати структуру договорів і забезпечити більш контрольований процес створення документів. Запропонована система має модульну структуру, що створює основу для подальшого розширення її функціональності, додавання нових шаблонів договорів, розширення правил перевірки та інтеграції з іншими інформаційними ресурсами.

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
						64
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. ISO/IEC 27001:2022. Information security, cybersecurity and privacy protection – Information security management systems – Requirements. International Organization for Standardization. 2022. URL: <https://www.iso.org/standard/27001> (дата звернення: 24.02.2026).

2. ISO/IEC 27002:2022. Information security, cybersecurity and privacy protection – Information security controls. International Organization for Standardization. 2022. URL: <https://www.iso.org/standard/75652.html> (дата звернення: 24.02.2026).

3. NIST Special Publication 800-53 Rev. 5. Security and Privacy Controls for Information Systems and Organizations. National Institute of Standards and Technology. 2020. URL: <https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final> (дата звернення: 24.02.2026).

4. NIST Special Publication 800-218. Secure Software Development Framework Version 1.1 / National Institute of Standards and Technology. 2022. URL: <https://csrc.nist.gov/publications/detail/sp/800-218/final> (дата звернення: 24.02.2026).

5. OWASP. OWASP Top 10 – 2021: The Ten Most Critical Web Application Security Risks. 2021. URL: <https://owasp.org/Top10/> (дата звернення: 24.02.2026).

6. OWASP. Application Security Verification Standard 4.0.3. 2019. URL: <https://owasp.org/www-project-application-security-verification-standard/> (дата звернення: 24.02.2026).

7. Fielding R., Nottingham M., Reschke J. RFC 9110: HTTP Semantics. 2022. URL: <https://www.rfc-editor.org/rfc/rfc9110.html> (дата звернення: 24.02.2026).

8. OpenAPI Initiative. OpenAPI Specification 3.1.0 Released. 2021. URL: <https://www.openapis.org/blog/2021/02/18/openapi-specification-3-1-released> (дата звернення: 24.02.2026).

9. OpenAPI Initiative. OpenAPI Specification – Version 3.1.0. 2021. URL: <https://swagger.io/specification/> (дата звернення: 24.02.2026).

					КВРІСТ.230156.23.01.05 ПЗ	Арк.
						65
Зм.	Арк.	№ докум.	Підпис	Дата		

10. JSON Schema. Draft 2020-12. 2020. URL: <https://json-schema.org/draft/2020-12> (дата звернення: 24.02.2026).

11. Directive 2022/2555 of the European Parliament and of the Council of 14 December 2022. 2022. URL: <https://eur-lex.europa.eu/eli/dir/2022/2555/oj/eng> (дата звернення: 24.02.2026).

12. Regulation 2024/1689 of the European Parliament and of the Council of 13 June 2024. 2024. URL: <https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng> (дата звернення: 24.02.2026).

13. Regulation 2024/1183 of the European Parliament and of the Council of 11 April 2024. 2024. URL: <https://eur-lex.europa.eu/eli/reg/2024/1183/oj/eng> (дата звернення: 24.02.2026).

14. Про затвердження Порядку проведення процедури оцінки відповідності у сферах електронної ідентифікації та електронних довірчих послуг : Постанова Кабінету Міністрів України; Порядок від 13.09.2024 № 1062 URL: <https://zakon.rada.gov.ua/go/1062-2024-%D0%BF> (дата звернення: 24.02.2026).

15. Міністерство цифрової трансформації України. Скільки паперу зекономили українці в 2021 році, скориставшись е-підписом понад 8,7 млрд разів. 2022. URL: <https://thedigital.gov.ua/news/technologies/skilki-paperu-zekonomili-ukraintsi-v-2021-rotsi-skoristavshis-e-pidpisom-ponad-87-mlrd-raziv-zvit> (дата звернення: 24.02.2026).

16. Дія. Як отримати електронний підпис?. 2021. URL: <https://diia.gov.ua/faq/1> (дата звернення: 24.02.2026).

17. Дія. Дія.Підпис – Кваліфікований електронний підпис. 2021. URL: https://ca.diia.gov.ua/faq_diia_id (дата звернення: 24.02.2026).

18. UNCITRAL. UNCITRAL Model Law on Automated Contracting. 2024. URL: <https://uncitral.un.org/en/mlac> (дата звернення: 24.02.2026).

					КвРІСТ.230156.23.01.05 ПЗ	Арк.
						66
Зм.	Арк.	№ докум.	Підпис	Дата		

19. UNCITRAL. UNCITRAL Model Law on Automated Contracting: Text \ 2024. URL: https://uncitral.un.org/sites/uncitral.un.org/files/mlac_en.pdf (дата звернення: 24.02.2026).

20. UNCITRAL. Guide to Enactment of the UNCITRAL Model Law on Automated Contracting. 2024. URL: <https://uncitral.un.org/sites/uncitral.un.org/files/2424674e-mlautomatedcontracting-ebook.pdf> (дата звернення: 24.02.2026).

21. Lam K.-Y., Cheng V. C. W., Yeong Z. K. Applying Large Language Models for Enhancing Contract Drafting. 2023. URL: <https://ceur-ws.org/Vol-3423/paper7.pdf> (дата звернення: 24.02.2026).

22. Narendra S., Shetty K., Ratnaparkhi A. *Enhancing Contract Negotiations with LLM-Based Legal Document Comparison*. Proceedings of the Natural Language Processing Workshop 2024. 2024. URL: <https://aclanthology.org/2024.nllp-1.11.pdf> (дата звернення: 24.02.2026).

23. Xu W., Zhao W., et al. *ConReader: Exploring Implicit Relations in Contracts for Contract Clause Extraction*. Proceedings of EMNLP 2022. 2022. URL: <https://aclanthology.org/2022.emnlp-main.166.pdf> (дата звернення: 24.02.2026).

24. Carneiro-Diaz V., et al. Automated legal analysis of rental contract clauses using large language models. *SoftwareX*. 2025. URL: <https://www.sciencedirect.com/science/article/pii/S2352711025003036> (дата звернення: 24.02.2026).

25. Stathis G. Explainable Large Language Models & iContracts. 2024. URL: <https://www.scitepress.org/Papers/2024/126074/126074.pdf> (дата звернення: 24.02.2026).

26. Aejas B., et al. Efficient legal contract clause extraction using a QA-based knowledge distillation approach. 2025. DOI: 10.1007/s11280-025-01375-7. URL: <https://dl.acm.org/doi/abs/10.1007/s11280-025-01375-7> (дата звернення: 24.02.2026).

27. Galassi A., et al. *Unfair clause detection in terms of service across multiple languages*. *AI and Law*. 2025. URL: <https://link.springer.com/article/10.1007/s10506-024-09398-7> (дата звернення: 24.02.2026).

					КВРІСТ.230156.23.01.05 ПЗ	Арк.
						67
Зм.	Арк.	№ докум.	Підпис	Дата		

28. Niklaus J., et al. *Online Terms of Service Dataset*. Hugging Face Datasets. 2021. URL: https://huggingface.co/datasets/joelniklaus/online_terms_of_service (дата звернення: 24.02.2026).

29. *ContractEval: Benchmarking LLMs for Clause-Level Legal Risk Identification in Commercial Contracts*. arXiv. 2025. URL: <https://arxiv.org/html/2508.03080v1> (дата звернення: 24.02.2026).

30. European Commission. Discussion paper on automated contracting. 2025. URL: https://commission.europa.eu/document/download/6b8c3d6e-ef3e-4cb9-8b3f-ad20f518e585_en?filename=Discussion+paper+on+automated+contracting+-final.pdf (дата звернення: 24.02.2026).

31. DocuSign. The State of Contract Management. 2020. URL: <https://betterbusiness-fbnz.fujifilm.com/wp-content/uploads/2022/02/DocuSign-CLM-State-Of-Contract-2020-Report-FUJIFILM-Process-Automation.pdf> (дата звернення: 24.02.2026).

32. World Commerce & Contracting. Contract Lifecycle Management: from technology to transformation. 2024. URL: https://www.worldcc.com/Portals/IACCM/Files/11462_0_Contract%20Lifecycle%20Management.pdf (дата звернення: 24.02.2026).

33. FeaturedCustomers. Fall 2020 Customer Success Report: Contract Lifecycle Management. 2020. URL: https://cdn.featuredcustomers.com/customer_success_report/FC-CUSTOMER-SUCCESS-REPORT-FALL-2020-CONTRACT-LIFECYCLE-MANAGEMENT.pdf (дата звернення: 24.02.2026).

34. KPMG. Contract Lifecycle Management. 2020. URL: https://atlas.kpmg.com/docs/default-source/kpmg-use-case-navigator-documents/ucn601_0001_produkblatt-en.pdf (дата звернення: 24.02.2026).

35. PostgreSQL Global Development Group. PostgreSQL 16 Released!. 2023. URL: <https://www.postgresql.org/about/news/postgresql-16-released-2715/> (дата звернення: 24.02.2026).

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 68
Зм.	Арк.	№ докум.	Підпис	Дата		

36. PostgreSQL Global Development Group. PostgreSQL 16.0 Release Notes. 2023. URL: <https://www.postgresql.org/docs/release/16.0/> (дата звернення: 24.02.2026).

37. SQLite Consortium. SQLite Release 3.45.0 (2024-01-15). 2024. URL: https://sqlite.org/releaselog/3_45_0.html (дата звернення: 24.02.2026).

38. FastAPI. fastapi, версія 0.110.0. 2024. URL: <https://pypi.org/project/fastapi/> (дата звернення: 24.02.2026).

39. FastAPI. Tutorial – User Guide. 2024. URL: <https://fastapi.tiangolo.com/tutorial/> (дата звернення: 24.02.2026).

40. Pydantic. pydantic, версія 2.12.5. 2025. URL: <https://pypi.org/project/pydantic/> (дата звернення: 24.02.2026).

41. Pydantic. Releases: v2.12.5. GitHub. 2025. URL: <https://github.com/pydantic/pydantic/releases> (дата звернення: 24.02.2026).

42. Pallets Projects. Jinja2 3.1.3. 2024. URL: <https://pypi.org/project/Jinja2/3.1.3/> (дата звернення: 24.02.2026).

43. Pallets Projects. Jinja Documentation – Changes (3.1.x). 2022. URL: <https://jinja.palletsprojects.com/en/stable/changes/> (дата звернення: 24.02.2026).

44. python-docx. python-docx, версія 1.2.0. 2025. URL: <https://pypi.org/project/python-docx/> (дата звернення: 24.02.2026).

45. python-docx. python-docx 1.2.0 documentation. 2025. URL: <https://python-docx.readthedocs.io/> (дата звернення: 24.02.2026).

46. docxtpl. docxtpl, версія 0.20.2. 2025. URL: <https://pypi.org/project/docxtpl/> (дата звернення: 24.02.2026).

47. CourtBouillon. WeasyPrint v61 Has Been Released. 2024. URL: <https://www.courtbouillon.org/blog/00049-weasyprint-61/> (дата звернення: 24.02.2026).

48. WeasyPrint. WeasyPrint – The Awesome Document Factory. 2024. URL: <https://weasyprint.org/> (дата звернення: 24.02.2026).

					КвРІСТ.230156.23.01.05 ПЗ	Арк. 69
Зм.	Арк.	№ докум.	Підпис	Дата		

49. ReportLab. reportlab, версія 4.2.0. 2024. URL: <https://pypi.org/project/reportlab/> (дата звернення: 24.02.2026).

50. React Team. React 19. 2024. URL: <https://react.dev/blog/2024/12/05/react-19> (дата звернення: 24.02.2026).

51. DocuSign CLM. URL: <https://www.absyz.com/docusign-clm-the-ultimate-tool-for-managing-contracts/> (дата звернення: 24.02.2026).

52. ContractWorks. URL: <https://www.pcmag.com/reviews/contractworks> (дата звернення: 24.02.2026).

53. PandaDoc. URL: <https://www.pandadoc.com/blog/a-new-era-for-pandadoc-crafting-the-next-chapter-together/> (дата звернення: 24.02.2026).

					КВРІСТ.230156.23.01.05 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		70

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Назар МИХАЙЛОВ

Співавтор:

Назва: Система автоматизованої генерації та перевірки типових договорів на основі введених реквізитів та шаблонів

Експерт: Олег ВОЙЧУР

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 2.18%

Коефіцієнт подібності 2: 0.34%

Мікропробіли: 3

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2026-05-25 14:20:27.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2026-05-25

Дата



Доцент Андрій Нічепорук

експерт

Mon May 25 13:44:44 EEST 2026, Медзатий Дмитро Михайлович, Хмельницький національний університет, ХНУ

Anti-Plagiarism (<http://ap.km.ua>) v-15.701

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 7%

ID: 272221 Назва: КП Система автоматизованої генерації та перевірки типових договорів на основі введених реквізитів та шаблонів Додано в БД: 2026-05-25 Автора: Назар МИХАЙЛОВ Керівники: Олег ВОЙЧУР Консультанти: Олоненги:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	110613	814	2124 (2%)	25 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Михайлов Назар Олександрович

Тема: Система автоматизованої генерації та перевірки типових договорів на основі введених реквізитів та шаблонів

Спеціальність: 126 «Інформаційні системи та технології»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 64

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є проектування, реалізація та тестування інформаційної системи, призначеної для введення реквізитів, перевірки їх правильності, автоматичного заповнення шаблонів і формування готових договорів у зручному форматі.
2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.
3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: в першому розділі проаналізовано особливості підготовки договірної документації в сучасних умовах, визначено ключові проблеми ручних і напівавтоматизованих підходів та сформульовано вимоги до програмного рішення. В другому розділі розроблено загальну архітектуру інформаційної системи, структуру бази даних, а також алгоритм генерації та перевірки договорів. Обґрунтовано вибір сучасних програмних засобів, зокрема Python, FastAPI та реляційних баз даних. В третьому розділі виконано практичну реалізацію серверної та клієнтської частин системи, бази даних, механізму багаторівневої перевірки реквізитів та модуля генерації документів на основі шаблонів.
4. Позитивні сторони роботи: . створено працездатний програмний прототип, який забезпечує автоматизоване формування типових договорів, перевірку правильності введених даних та збереження сформованих документів. Це створює

передумови для підвищення швидкості підготовки документації та зменшення кількості помилок під час заповнення реквізитів.

5. Негативні сторони роботи: бажано було б розширити кількість підтримуваних шаблонів договорів для більш масштабного тестування системи в умовах реального документообігу та інтеграції з існуючими зовнішніми хмарними сервісами.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.


8. Інші зауваження: _____

9. Оцінка дипломної роботи: задовільно (D / 70)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Григорук О.М., доцент кафедри ІТІЗ

“ ” _____ 2026 р.

 _____ (підпис)

Зав. кафедри КІС
д-р. філософії Ользі ПАВЛОВІЙ

Михайлов Назар Олександрович

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи ІСТс-23-1

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

1 травня 2026 року



**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Назва кваліфікаційної роботи Система автоматизованої генерації та перевірки типових договорів на основі введених реквізитів та шаблонів

Автор Назар МИХАЙЛОВ

Освітня програма Інформаційні системи та технології

Рівень вищої освіти перший (бакалаврський)

Спеціальність 126 Інформаційні системи та технології

Науковий керівник: Войчур Олег Юрійович

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.
- 4) значна частина знайденого плагіату відноситься до списку використаних джерел


Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 2,18%; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

01.06.2026

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи


Підпис

Підпис

Підпис

Ольга ПАВЛОВА

Ім'я, ПРІЗВИЩЕ

Єлизавета ГНАТЧУК

Ім'я, ПРІЗВИЩЕ

Олег ВОЙЧУР

Ім'я, ПРІЗВИЩЕ