

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавр

на тему: **«Розроблення вебдодатку для накладення підпису на документи»**

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи
спеціальності 126 Інформаційні
системи та технології
ступеня вищої освіти бакалавр
групи 126ІСТ_бд_2022
Кравчук Станіслав Русланович
Керівник: Протас Надія Михайлівна
Рецензент: Муравльов Володимир
Вячеславович

Полтава – 2026 року

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

Освітня програма Інформаційні управляючі системи
Спеціальність 126 Інформаційні системи та технології
Рівень вищої освіти перший (бакалаврський)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ **Юрій УТКІН**
«10» червня 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Кравчуку Станіславу Руслановичу

1. Тема роботи:
«Розроблення вебдодатку для накладення підпису на документи»,
керівник роботи к.с.-г.н., доцент, доцент кафедри інформаційних систем та технологій Протас Надія Михайлівна.
Затверджено наказом закладу вищої освіти від «10» квітня 2026 року № 383 ст.
2. Строк подання здобувачем вищої освіти роботи «08» червня 2026 року.
3. Вихідні дані до роботи: законодавство України у сфері електронного документообігу, ДСТУ 4145-2002, ГОСТ 34.602-89, дані офіційних сайтів <https://www.php.net/>, <https://www.mysql.com/>, <https://mozilla.github.io/pdf.js/>, <https://developer.mozilla.org/> (для JavaScript та Canvas API), середовища прикладних програм phpMyAdmin, OpenServer 6.5.1, Visual Studio Code.
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

Розділ 1. Аналіз предметної області систем підписання документів та огляд існуючих рішень
Розділ 2. Проектування вебдодатку для накладення підпису на документи
Розділ 3. Програмна реалізація та оцінювання вебдодатку
5. Перелік графічного матеріалу: схеми, рисунки, графіки, діаграми за темою та об'єктом дослідження.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання отримав
Оцінювання економічної ефективності результатів дослідження	Дивнич О. Д., к. е. н., доцент, доцент кафедри економіки та публічного управління	01.04.2026 р.	29.05.2026 р.

7. Дата видачі завдання «10» червня 2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Вибір і затвердження теми роботи	02.06.2025 р. – 04.06.2025	
2	Складання і затвердження розгорнутого плану та завдання на кваліфікаційну роботу	05.06.2025 р. – 10.06.2025 р.	
3	Опрацювання джерел інформації	11.06.2025 р. – 15.06.2025 р.	
4	Збір, вивчення і обробка інформації, необхідної для виконання роботи	16.06.2025 р. – 20.06.2025 р.	
5	Виконання теоретико-методологічного розділу роботи	08.09.2025 р. – 01.11.2025 р.	
6	Виконання дослідницько-аналітичного розділу роботи	19.01.2026 р. – 14.02.2026 р.	
7	Виконання проєктно-рекомендаційного розділу роботи	16.02.2026 р. – 31.03.2026 р.	
8	Оцінювання економічної ефективності результатів дослідження	01.04.2026 р. – 29.05.2026 р.	
9	Оформлення тексту роботи	01.06.2026 р. – 06.06.2026р.	
10	Попередній захист роботи на кафедрі	08.06.2026 р.	
11	Доопрацювання роботи з урахуванням зауважень і пропозицій	09.06.2026 р. – 10.06.2026 р.	
12	Нормоконтроль	11.06.2026 р. – 15.06.2026 р.	
13	Захист кваліфікаційної роботи	16.06.2026 р. - 18.06.2026 р.	

Здобувач вищої освіти

Станіслав КРАВЧУК

Керівник роботи

Надія ПРОТАС

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ СИСТЕМ ПІДПИСАННЯ ДОКУМЕНТІВ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	9
1.1 Поняття електронного підпису та графічного накладення підпису на документ	9
1.2 Нормативно-правове регулювання електронного документообігу в Україні	12
1.3 Аналіз існуючих систем підписання документів та обґрунтування розробки	15
1.4 Методи і засоби розроблення вебдодатків на PHP	18
РОЗДІЛ 2 ПРОЄКТУВАННЯ ВЕБДОДАТКУ ДЛЯ НАКЛАДЕННЯ ПІДПISY НА ДОКУМЕНТИ	23
2.1 Технічне завдання на розроблення вебдодатку	23
2.2 Проєктування бази даних та архітектури застосунку	26
2.3 Проєктування інтерфейсу користувача	29
2.4 Моделювання процесів за допомогою UML діаграм системи	32
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ОЦІНЮВАННЯ ВЕБДОДАТКУ ..	36
3.1 Реалізація серверної та клієнтської частин вебдодатку	36
3.2 Розгортання та тестування вебдодатку в середовищі OpenServer	39
3.3 Керівництво користувача вебдодатку	42
3.4 Техніко-економічне обґрунтування ефективності розробленої системи	46
ВИСНОВКИ	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	52
ДОДАТКИ	55

ВСТУП

Актуальність. У сучасному світі електронний документообіг стає невід'ємною складовою як державного управління, так і приватного бізнесу. За даними аналітичних досліджень [1, 13], у 2024 році обсяг ринку систем електронного документообігу оцінювався приблизно у 20 млн доларів США, а середньорічний темп зростання за 2020-2024 роки становить 19%. Запровадження обов'язкового електронного документообігу для всіх державних установ України у 2024 році додатково прискорило попит на інструменти цифрового підписання документів. Паперове підписання договорів, актів і заяв потребує фізичної присутності сторін, витрат на друк, сканування та кур'єрську доставку, що суттєво уповільнює бізнес-процеси та підвищує операційні витрати організацій [2, 7].

Незважаючи на наявність на ринку комерційних хмарних рішень – Вчасно.ЕДО, Deals Sign, DocuSign – вони мають низку обмежень для малих організацій та навчальних установ: відсутність безкоштовних повнофункціональних тарифів, обов'язкова передача конфіденційних документів на зовнішні сервери, залежність від постійного доступу до мережі Інтернет та неможливість локального розгортання [1, 4, 16]. Це формує потребу у доступному вебдодатку з відкритою архітектурою, що розгортається локально, забезпечує графічне накладення підпису на PDF-документи та не вимагає інтеграції з зовнішніми центрами сертифікації ключів.

Метою кваліфікаційної роботи є проєктування та розробка вебдодатку для накладення підпису на документи з використанням сучасних вебтехнологій, що забезпечує автоматизацію процесу підписання PDF-документів, розмежування доступу за ролями користувачів та локальне розгортання в середовищі OpenServer.

Відповідно до мети роботи необхідно вирішити наступні *завдання*:

1. Провести аналіз предметної галузі, розмежувати поняття кваліфікованого електронного підпису та графічного накладення підпису,

дослідити нормативно-правову базу електронного документообігу в Україні та здійснити огляд існуючих систем підписання документів;

2. Спроекувати архітектуру бази даних для зберігання даних про користувачів, документи, запити підписання, накладені підписи та журнал дій, а також розробити технічне завдання, прототипи інтерфейсу та UML-діаграми системи;

3. Реалізувати вебдодаток, включаючи серверну частину на PHP, клієнтський інтерфейс з підтримкою PDF.js та Canvas API, розгорнути систему в OpenServer, провести функціональне тестування та виконати техніко-економічне обґрунтування ефективності розробки.

Об'єкт дослідження – процеси проєктування та розробки вебдодатків для автоматизації документообігу з підтримкою графічного накладення підпису на PDF-документи.

Предмет дослідження – проєктування та розробка вебдодатку для накладення підпису на документи з використанням PHP, MySQL/MariaDB, PDF.js та Canvas API у середовищі локального розгортання OpenServer.

Методи досліджень – дослідження базуються на методах програмної інженерії, аналізу вимог до інформаційних систем, проєктування реляційних баз даних (ER-діаграми, нормалізація до 3НФ), розробки вебдодатків за архітектурним патерном MVC, тестування за методом «чорної скриньки», а також техніко-економічного аналізу на основі методик TCO та ROI.

Інформаційна база – чинне законодавство України у сфері електронного документообігу, нормативні документи (ДСТУ 4145-2002, ГОСТ 34.602-89), наукові статті та аналітичні матеріали про ринок електронного документообігу, технічна документація до технологій PHP 8.x, MySQL/MariaDB, PDF.js, Canvas API, а також інтернет-ресурси щодо розробки вебдодатків і стандартів безпеки інформаційних систем.

Практична значущість – розроблений вебдодаток забезпечує повний цикл підписання PDF-документів у межах локальної мережі організації без залежності від зовнішніх хмарних сервісів. Реалізована система розмежування доступу за

трьома ролями, двосторонній інтерфейс підписання з підтримкою малювання підпису мишею та завантаження готового зображення, а також журналювання всіх дій роблять систему придатною для впровадження у малих підприємствах, навчальних закладах та державних установах з обмеженим бюджетом.

Результати роботи апробовані в рамках наукової конференції здобувачів вищої освіти за результатами науково-дослідної роботи у 2025-2026 рр.

Структура кваліфікаційної роботи логічно пов'язана з задачами досліджень і містить перелік умовних позначень, вступ, три розділи основної частини, висновки, список використаних джерел. Загальний обсяг текстової частини кваліфікаційної роботи складає 54 сторінки формату А4. Вона містить 14 рисунків і 12 таблиць.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ СИСТЕМ ПІДПИСАННЯ ДОКУМЕНТІВ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Поняття електронного підпису та графічного накладення підпису на документ

З розвитком інформаційних технологій та прискоренням процесів цифрової трансформації в Україні електронний документообіг стає невіддільною частиною як бізнес-середовища, так і взаємодії громадян із державними установами. У 2024 році впровадження електронного документообігу стало обов'язковим для всіх державних установ, що дозволило скоротити час на обробку документації та зменшити корупційні ризики. Одним із ключових елементів цієї системи є механізм засвідчення документів – підписання. Проте у практичному застосуванні поняття «підписання документів» охоплює принципово різні за своєю природою підходи, які важливо чітко розмежовувати.

Електронний підпис у правовому розумінні – це криптографічний механізм, що забезпечує юридичну значущість електронного документа. Кваліфікований електронний підпис (КЕП) – це вид електронного підпису, отриманого за результатом криптографічного перетворення набору електронних даних, який додається до цього набору або логічно з ним поєднується і дає змогу підтвердити його цілісність та ідентифікувати підписувача. КЕП накладається за допомогою особистого ключа та перевіряється за допомогою відкритого ключа. Відповідно до статті 18 Закону України «Про електронні довірчі послуги» (№2155 від 7 листопада 2018 року), саме КЕП прирівнюється до особистого підпису. Станом на сьогодні цей закон отримав оновлену назву – «Про електронну ідентифікацію та електронні довірчі послуги», що відображає розширення його сфери регулювання [4].

Математичну основу КЕП складає алгоритм асиметричного шифрування, зокрема схема цифрового підпису. У спрощеному вигляді операція перевірки підпису може бути записана як:

$$V(pk, m, \sigma) = \begin{cases} 1, \text{ при } \sigma \text{ є дійсним для } m; \\ 0, \text{ інакше;} \end{cases} \quad (1.1)$$

де pk – відкритий ключ підписанта,
 m – електронний документ (або його геш),
 σ – електронний підпис,
 V – функція верифікації [3].

Графічне (факсимільне) накладення підпису – це принципово інший підхід, що не є криптографічним. У господарській практиці факсиміле – це відтворений за допомогою певних засобів копіювання оригінальний власноручний підпис особи, мета якого полягає у спрощенні роботи особи, якій потрібно підписувати велику кількість документів. У контексті вебдодатків графічне накладення підпису передбачає розміщення растрового зображення підпису (або підпису, намальованого мишею чи стилусом) у визначеному місці сторінки PDF-документа. Такий підпис є лише візуальним атрибутом і сам по собі не гарантує ні цілісності документа, ні криптографічної ідентифікації підписанта. На рисунку 1.1 наведена концептуальна схема, що ілюструє відмінності між двома механізмами підписання документів.



Рисунок 1.1 – Порівняльна схема електронного підпису та графічного накладення підпису

Для кращого розуміння відмінностей між двома підходами наведено порівняльну таблицю (таблиця 1.1).

Таблиця 1.1 – Порівняння електронного підпису (КЕП) та графічного накладення підпису

Критерій	Кваліфікований електронний підпис (КЕП)	Графічне накладення підпису
Правова основа	Закон України №2155, регламент eIDAS	Цивільний кодекс України, внутрішні угоди сторін
Технічна природа	Криптографічне перетворення даних	Розміщення растрового зображення
Засіб реалізації	Токен, смарт-картка, хмарний КЕП	Завантажене зображення, Canvas API
Гарантія цілісності	Так (будь-яка зміна документа робить підпис недійсним)	Ні
Ідентифікація підписанта	Криптографічна (через сертифікат АЦСК)	Візуальна (за зовнішнім виглядом)
Юридична сила	Прирівнюється до власноручного підпису	Потребує письмової угоди сторін
Сфера застосування	Держреєстрація, податкова звітність, публічні закупівлі	Внутрішній документообіг, узгодження, попереднє підписання

Як видно з таблиці 1.1, обидва підходи мають свою область застосування і не є взаємовиключними. Використання факсимільного відтворення підпису допускається у випадках, встановлених законом або за письмовою згодою сторін, у якій мають міститися зразки відповідного аналога власноручних підписів. Це означає, що графічне накладення підпису є цілком легітимним інструментом, проте його правова сила залежить від контексту та домовленостей між сторонами [5].

У рамках розроблюваного вебдодатку реалізується саме графічний підхід до підписання – накладення зображення підпису на PDF-документ засобами браузера. Такий підхід є достатнім для організації внутрішнього документообігу підприємства, де підписанти вже ідентифіковані системою через механізм аутентифікації, а юридична сила документів забезпечується внутрішніми процедурами організації. Електронний документообіг сьогодні є невід'ємною

складовою ефективною та стабільною роботи підприємства, і якщо раніше він охоплював лише бухгалтерські документи, то зараз – це весь спектр документів в електронному вигляді. Розроблений вебдодаток спрямований саме на задоволення цієї зростаючої потреби у простому та доступному інструменті внутрішнього підписання документів [6].

1.2 Нормативно-правове регулювання електронного документообігу в Україні

Правова база електронного документообігу в Україні формувалася поступово і сьогодні являє собою розгалужену систему законодавчих та підзаконних актів. Основу цієї системи складають два ключові закони, прийнятих ще на початку 2000-х років, однак неодноразово оновлених відповідно до вимог часу та євроінтеграційних прагнень держави. Відносини, пов'язані з електронним документообігом та використанням електронних документів, регулюються Конституцією України, Цивільним кодексом України, законами України «Про інформацію», «Про захист інформації в інформаційно-комунікаційних системах», «Про електронні комунікації», а також іншими нормативно-правовими актами.

Базовим документом є Закон України «Про електронні документи та електронний документообіг» від 22.05.2003 р. № 851-IV. Цей Закон встановлює основні організаційно-правові засади електронного документообігу та визначає електронний документ як документ, інформація в якому зафіксована у вигляді електронних даних, включаючи обов'язкові реквізити документа. До обов'язкових реквізитів електронного документа, зокрема, відносяться: назва документа, дата складання, найменування підприємства або особи, зміст господарської операції, а також підпис уповноваженої особи [8]. Закон неодноразово оновлювався, останні зміни внесені у грудні 2022 року [9].

Другим ключовим нормативним актом є Закон України «Про електронну ідентифікацію та електронні довірчі послуги» від 05.10.2017 р. № 2155-VIII (у початковій редакції – «Про електронні довірчі послуги»). Цим законом врегульовуються відносини у сферах надання електронних довірчих послуг та електронної ідентифікації, а одним із найважливіших його положень є взаємне визнання українських та іноземних сертифікатів відкритих ключів та електронних підписів. Станом на сьогодні закон діє в редакції від 18.12.2024 р., що свідчить про його активну актуалізацію у зв'язку із євроінтеграційними процесами [10].

З технічного боку реалізацію електронного підпису в Україні регламентує ДСТУ 4145-2002 «Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевірка». Цей стандарт встановлює механізм цифрового підписування, що ґрунтується на властивостях груп точок еліптичних кривих над полями, та правила застосування цього механізму до повідомлень, що їх пересилають каналами зв'язку та/або обробляють у комп'ютеризованих системах. Згідно з наказом Мінцифри від 30 жовтня 2020 року № 140/614, після 1 січня 2021 року стандарт використовується для створення кваліфікованого електронного підпису разом із функцією гешування «Купина». Алгоритм формування підпису на еліптичних кривих над полем $GF(2^m)$ описується виразом [11]:

$$S = (e + \alpha \cdot r) \cdot d^{-1}(\text{mod } n), \quad (1.2)$$

де e – геш-значення документа,
 r – координата точки на кривій,
 d – особистий ключ підписанта,
 n – порядок базової точки,
 S – значення підпису.

На рисунку 1.2 наведено структуру нормативно-правового забезпечення електронного документообігу в Україні.



Рисунок 1.2 – Структура нормативно-правового забезпечення електронного документообігу в Україні

Важливим елементом системи регулювання є інститут кваліфікованих надавачів електронних довірчих послуг (КНЕДП) – акредитованих центрів сертифікації ключів (АЦСК). До найвідоміших КНЕДП в Україні належать: АЦСК «Дія» (КНЕДП ДПС), АЦСК «Приватбанк», АЦСК «УкрСиббанк», «Укрзалізниця» та ін. З 01.01.2024 отримати послуги з формування кваліфікованих сертифікатів відкритих ключів за довіреністю неможливо – ідентифікація фізичної особи здійснюється виключно за умови її особистої присутності.

Паралельно Україна активно адаптує своє законодавство до вимог Європейського Союзу, зокрема до регламенту eIDAS (EU) No 910/2014, що встановлює єдині правила електронної ідентифікації та довірчих послуг на території ЄС. Дослідники підкреслюють, що правові засади розвитку електронного урядування в Україні в умовах європейської інтеграції потребують адаптації нормативно-правового середовища до міжнародних стандартів, що забезпечують юридичну силу електронних документів та захист персональних даних. Основні нормативні акти у сфері електронного документообігу та їх ключові положення систематизовано в таблиці 1.2.

Таблиця 1.2 – Основні нормативно-правові акти у сфері електронного документообігу в Україні

№	Нормативний акт	Рік прийняття (остання редакція)	Ключові положення
1	ЗУ «Про електронні документи та електронний документообіг» № 851-IV	2003 (2022)	Поняття електронного документа, засади ЕДО
2	ЗУ «Про електронну ідентифікацію та електронні довірчі послуги» № 2155-VIII	2017 (2024)	КЕП, АЦСК, довірчі послуги, eIDAS-сумісність
3	ДСТУ 4145-2002	2002 (2021)	Алгоритм формування та перевірки ЦП на еліптичних кривих
4	ДСТУ 7564:2014 «Функція гешування Купина»	2014 (2021)	Хеш-функція для використання з КЕП
5	Постанова КМУ № 992 від 07.11.2018	2018	Вимоги у сфері електронних довірчих послуг

Таким чином, нормативно-правова база України у сфері електронного документообігу є достатньо розвиненою та продовжує вдосконалюватись відповідно до євроінтеграційних зобов'язань. Розробка вебдодатків у цій сфері має спиратися на чинне законодавство, зокрема в частині розмежування видів підписів та їх правового статусу. У рамках даної роботи реалізується функціонал графічного накладення підпису, що відповідає положенням Цивільного кодексу України щодо допустимості факсимільних підписів за умови відповідного внутрішнього регулювання.

1.3 Аналіз існуючих систем підписання документів та обґрунтування розробки

Ринок програмних рішень для електронного документообігу та підписання документів в Україні активно розвивається. За даними YouControl, у 2024 році обсяг українського сегменту ринку ЕДО оцінювався приблизно у 20 млн доларів, а середньорічний темп зростання за період 2020-2024 років становить 19%. На

ринку присутні як вітчизняні, так і зарубіжні рішення, що відрізняються за функціональністю, ціновою моделлю, механізмом підписання та сферою застосування. Розглянемо найбільш поширені з них.

Вчасно.ЕДО – провідна українська хмарна платформа електронного документообігу. Сервіс надає можливості для створення, погодження, підписання, обміну та зберігання документів онлайн і розрахований на всіх учасників компанії – від директора до працівника складу. Платформі довіряють понад 700 тис. українських компаній. Серед переваг – наявність безкоштовного тарифу та дисконтна система; серед недоліків – мобільний застосунок працює лише з власними ключами Вчасно або Дія.Підписом.

Deals Sign – українське онлайн-рішення для погодження та підписання юридично значущих документів. Система орієнтована на директорів, юристів, бухгалтерів та IT-керівників організацій будь-якого розміру, включаючи держустанови. Функціонал включає як класичний набір (візування, надсилання контрагентам, шаблони), так і можливість інтеграції з кадровими, ERP, CRM та іншими системами через REST API.

DocuSign – міжнародний лідер ринку електронного підписання. Ідентифікація підписанта може забезпечуватися кількарівневою аутентифікацією за допомогою електронної пошти, коду доступу, СМС-повідомлення або біометричних даних, а кожен підписаний документ має сертифікат про завершення як доказ підписання сторонами. Втім, вартість підписки та відсутність інтеграції з вітчизняними КНЕДП обмежують його застосування в українських реаліях.

Signy – вітчизняний сервіс, що підтримує підписання із застосуванням КЕП (файлового або токена), а також Mobile ID. Серед можливостей – налаштування персонального маршруту підписання, відстеження зміни статусу документа онлайн, редагування та коментування документів, завантаження документів разом із підписами та протоколом підписання для підтвердження юридичної значущості.

Порівняльний аналіз розглянутих систем за ключовими критеріями наведено в таблиці 1.3.

Таблиця 1.3 – Порівняльний аналіз існуючих систем підписання документів

Критерій	Вчасно.ЕДО	Deals Sign	DocuSign	Signy
Походження	Україна	Україна	США	Україна
Тип підпису	КЕП, Дія.Підпис	КЕП, графічний	Графічний, цифровий	КЕП, Mobile ID
Безкоштовний план	Так (обмежений)	Ні	Ні	Обмежений
Інтеграція з 1C/ERP	Так	Так (REST API)	Ні	Часткова
Хмарне сховище	Так (Amazon S3)	Так	Так	Так
Відповідність ЗУ №2155	Повна	Повна	Часткова	Повна
Локальне розгортання	Ні	Ні	Ні	Ні
Відкритий вихідний код	Ні	Ні	Ні	Ні

З таблиці 1.3 видно, що всі розглянуті системи є хмарними рішеннями без можливості локального розгортання і мають закритий вихідний код, що суттєво обмежує їх використання в організаціях з особливими вимогами до конфіденційності даних або обмеженим доступом до мережі Інтернет. Крім того, жодна із систем не є доступною для безкоштовного вивчення та адаптації в навчальних або дослідницьких цілях.

Виявлені недоліки існуючих рішень та наявна потреба в освоєнні технологій підписання документів обґрунтовують доцільність розробки власного вебдодатку. Показник покриття потреби C у відсотках від загальної кількості функціональних вимог n до системи описується виразом:

$$C = \frac{f_{impl}}{n} \times 100\%, \quad (1.3)$$

де f_{impl} – кількість реалізованих функцій,

n – загальна кількість визначених функціональних вимог.

Для навчального вебдодатку достатньо реалізувати базовий набір функцій, що забезпечить $C \geq 70\%$ [13]. На рисунку 1.3 наведено узагальнену схему типового процесу підписання документа в сучасній системі ЕДО.



Рисунок 1.3 – Типовий процес підписання документа в системі ЕДО

Розроблюваний вебдодаток відрізняється від аналогів такими особливостями: локальне розгортання в середовищі OpenServer, відкрита архітектура на базі PHP та MySQL, орієнтація на графічне накладення підпису без залежності від зовнішніх КНЕДП, а також доступність для адаптації та розширення. Такий підхід відповідає задачам внутрішнього документообігу малого підприємства або навчальної установи, де розгортання хмарних корпоративних рішень є економічно невиправданим або технічно неможливим.

1.4 Методи і засоби розроблення вебдодатків на PHP

Вибір методу розробки та технологічного стеку є одним з найважливіших рішень перед початком будь-якого програмного проєкту. Від правильності цього

вибору залежить ефективність процесу розробки, якість кінцевого продукту та витрати на його підтримку. У розробці вебзастосунків сьогодні домінують два методологічні підходи – каскадний (Waterfall) та гнучкий (Agile).

Каскадна модель (Waterfall) передбачає суворо послідовне проходження всіх фаз: аналіз вимог, проєктування, реалізація, тестування, впровадження. Методологія Waterfall передбачає повне завершення попередньої фази перед початком наступної, а завдяки тому, що більшість досліджень проводиться заздалегідь, можна точно оцінити витрати часу, необхідного для виконання кожної вимоги. Водночас її суттєвим недоліком є відсутність гнучкості – будь-яка зміна вимог потребує перегляду вже завершених фаз. Методологія Agile є протилежністю Waterfall: вона заснована на ітераційній розробці, коли продукт вдосконалюється у коротких циклах (спринтах). За даними Project Management Institute, 71% організацій повідомляють про більшу гнучкість управління проєктами завдяки використанню гнучких методологій, що сприяє підвищенню задоволеності клієнтів і скороченню термінів доставки. Для розробки розглядуваного вебдодатку обрано методологію Agile, оскільки вона дозволяє вносити зміни на будь-якому етапі та отримувати проміжні робочі версії продукту [17].

Ключовим елементом технологічного стеку є мова програмування серверної частини. PHP у 2024 році залишається однією з найпоширеніших мов програмування для веброзробки: приблизно 75% усіх сайтів у світі все ще використовують PHP у тому чи іншому вигляді; вихід версій PHP 8.x суттєво підвищив продуктивність і безпеку мови, а також додав підтримку сучасних стандартів, таких як типізація та синтаксичні нововведення. PHP є оптимальним вибором для реалізації серверної логіки вебдодатку підписання документів, оскільки забезпечує просту інтеграцію з реляційними СУБД, підтримує роботу з файлами та зображеннями «з коробки», а також не потребує додаткового налаштування середовища виконання у складі OpenServer [18].

Для зберігання даних (облікові записи користувачів, метадані документів, журнал підписань) у розроблюваному застосунку використовується MySQL –

одна зі СУБД, що входить до складу OpenServer 6.5.1. Переваги MySQL включають високу швидкість читання, стабільність, велику екосистему, простоту адміністрування та підтримку більшості хостингів; у зв'язці з PHP вона часто використовується для розробки корпоративних сайтів і e-commerce-проектів. Загальний технологічний стек застосунку, що розробляється, наведено на рисунку 1.4.

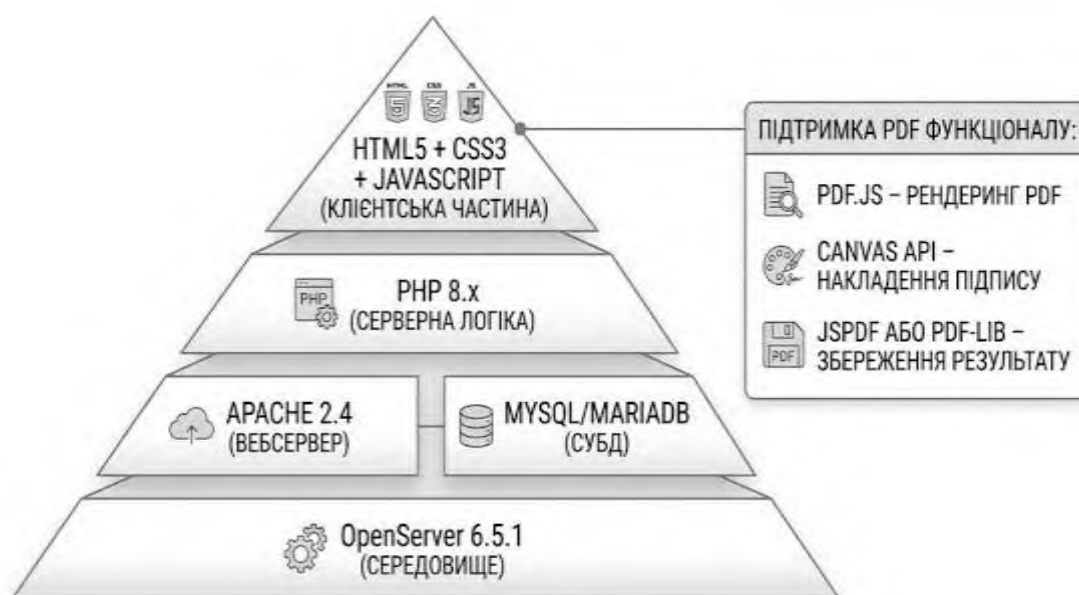


Рисунок 1.4 – Технологічний стек розроблюваного вебдодатку

На клієнтській стороні для рендерингу PDF-документів застосовується бібліотека PDF.js від Mozilla. PDF.js використовує JavaScript для рендерингу PDF у HTML5-елементи canvas без плагінів та реалізує трирівневу архітектуру: ядро (розбір PDF), API відображення та інтерфейс переглядача. Накладення зображення підпису здійснюється засобами Canvas API браузера: зображення підпису малюється поверх рендеру сторінки PDF у визначеній користувачем позиції. Для збереження результату – PDF з накладеним підписом – застосовується бібліотека pdf-lib або jsPDF, що дозволяють вбудувати растрове зображення безпосередньо в структуру PDF-файлу [19]. Порівняльну характеристику основних клієнтських та серверних технологій, задіяних у розробці, наведено в таблиці 1.4.

Таблиця 1.4 – Технологічні засоби розроблення вебдодатку та їх призначення

Технологія	Тип	Версія	Призначення у проєкті
PHP	Мова програмування	8.x	Серверна логіка, робота з файлами, сесії
MySQL/MariaDB	СУБД	10.x	Зберігання даних користувачів і документів
Apache	Вебсервер	2.4	Обробка HTTP-запитів
HTML5 / CSS3	Розмітка / стилі	–	Інтерфейс користувача
JavaScript	Мова програмування	ES2022+	Клієнтська логіка
PDF.js	JS-бібліотека	4.x	Рендеринг PDF у браузері
Canvas API	Браузерний API	–	Накладення підпису на сторінку
pdf-lib	JS-бібліотека	1.x	Вбудовування зображення в PDF
OpenServer	Локальне середовище	6.5.1	Розгортання та тестування

Вибір PHP без стороннього фреймворку (або з мінімальним набором Composer-пакетів) обумовлений навчальним характером роботи та необхідністю повного розуміння архітектури застосунку. Розробник, що реалізує власний MVC-подібний шар маршрутизації та шаблонізатор, набуває глибоких знань про принципи роботи вебзастосунків, які є базою для подальшого вивчення фреймворків. Середовище OpenServer 6.5.1 забезпечує швидке налаштування локального сервера з Apache, PHP та MySQL у Windows, не потребуючи встановлення та тонкого налаштування кожного компонента окремо [20]. Час розгортання нового проєкту в OpenServer можна описати формулою:

$$T_{deploy} = T_{create_domain} + T_{import_db} + T_{config_php}, \quad (1.4)$$

де T_{create_domain} – час на створення локального домену (хвилини),

T_{import_db} – час на імпорт SQL-дампу,

T_{config_php} – час на налаштування параметрів PHP (наприклад, `upload_max_filesize`).

За практичним досвідом, сумарний час T_{deploy} для типового навчального проєкту не перевищує 10-15 хвилин [20]. Було проведено комплексний теоретичний аналіз предметної галузі: чітко розмежовано поняття кваліфікованого електронного підпису та графічного накладення підпису на документ, розглянуто нормативно-правову базу електронного документообігу в Україні з акцентом на актуальних змінах, здійснено порівняльний огляд існуючих вітчизняних та зарубіжних систем підписання документів. Аналіз показав відсутність на ринку доступного відкритого рішення з можливістю локального розгортання, що підтверджує доцільність розробки власного вебдодатку на основі стеку PHP + MySQL + Apache у середовищі OpenServer з реалізацією графічного накладення підпису засобами Canvas API та бібліотеки PDF.js.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ВЕБДОДАТКУ ДЛЯ НАКЛАДЕННЯ ПІДПISУ НА ДОКУМЕНТИ

2.1 Технічне завдання на розроблення вебдодатку

Технічне завдання (ТЗ) є основним вихідним документом для проєктування та розроблення будь-якої інформаційної системи. Відповідно до ГОСТ 34.602-89, ТЗ на автоматизовану систему є основним документом, що визначає вимоги та порядок створення системи, відповідно до якого проводиться розробка та її приймання при введенні в дію. Незважаючи на те, що зазначений стандарт є частково застарілим, він залишається чинним орієнтиром для оформлення технічної документації на програмні системи в Україні, зокрема у навчальному середовищі [21]. Нижче наведено технічне завдання на розроблення вебдодатку для накладення підпису на документи.

Найменування продукту розробки – «Вебдодаток для накладення підпису на документи». Замовник – кафедра інформаційних систем та технологій. Виконавець – здобувач вищої освіти навчального закладу. Планові терміни розробки встановлюються відповідно до графіка виконання бакалаврської роботи. Середовище розгортання – локальна Windows-машина з OpenServer 6.5.1 (Apache 2.4, PHP 8.x, MySQL/MariaDB).

Вебдодаток призначений для організацій та фізичних осіб, що потребують інструменту графічного накладення підпису на PDF-документи у рамках внутрішнього документообігу. Мета системи – спростити процес підписання документів шляхом надання вебінтерфейсу для завантаження PDF-файлів, розміщення зображення підпису у визначеному місці документа та збереження підписаного файлу.

Ролі користувачів системи визначають права доступу та функціональні можливості кожного учасника. В системі передбачено три ролі, описані в таблиці 2.1.

Таблиця 2.1 – Ролі користувачів вебдодатку та їх повноваження

Роль	Опис	Основні функції
Адміністратор	Керує системою в цілому	Управління користувачами, перегляд журналу дій, налаштування системи
Ініціатор підписання	Завантажує документи та призначає підписантів	Завантаження PDF, призначення підписантів, відстеження статусів
Підписант	Підписує призначені документи	Перегляд документа, накладення підпису, підтвердження або відхилення

На рисунку 2.1 наведено діаграму варіантів використання системи (Use Case), що відображає взаємодію між ролями та основними функціями.



Рисунок 2.1 – Діаграма варіантів використання вебдодатку

Функціональні вимоги визначають, що система повинна виконувати. Розроблюваний вебдодаток має забезпечувати такі функції: реєстрація та аутентифікація користувачів (логін і пароль із хешуванням засобами PHP); завантаження PDF-документів розміром до 20 МБ; перегляд завантаженого документа у браузері засобами PDF.js; завантаження або малювання зображення

підпису (Canvas API); позиціонування підпису на обраній сторінці документа за координатами; збереження підписаного PDF-файлу; зміна статусу документа (Завантажено → На підписанні → Підписано / Відхилено); перегляд особистого кабінету з переліком документів і їх статусами; адміністративна панель для управління користувачами та перегляду журналу дій [22].

Нефункціональні вимоги охоплюють аспекти безпеки, сумісності та продуктивності. З точки зору безпеки: паролі користувачів зберігаються у вигляді хеш-значень (функція `password_hash()` PHP з алгоритмом `PASSWORD_BCRYPT`); усі вхідні дані проходять валідацію та санітизацію для захисту від SQL-ін'єкцій і XSS-атак; файли документів зберігаються поза публічною директорією вебсервера. Безпечну систему документообігу можна визначити за такими факторами: підписання документів проводиться на пристрої клієнта, а сам ключ підпису та його паролі не передаються на сервер. З точки зору сумісності: застосунок має коректно відображатися у сучасних браузерях (Chrome, Firefox, Edge) без встановлення додаткових плагінів. Щодо продуктивності: час відповіді сервера на типовий запит не повинен перевищувати 2 секунди при локальному розгортанні [23].

Апаратні та програмні вимоги до середовища розгортання: процесор з тактовою частотою не менше 1 ГГц; оперативна пам'ять – не менше 4 ГБ; вільний простір на диску – не менше 2 ГБ; операційна система Windows 10/11; OpenServer 6.5.1 із модулями Apache 2.4, PHP 8.x, MySQL або MariaDB 10.x; сучасний веббраузер із підтримкою HTML5 Canvas API.

Показник повноти реалізації функціональних вимог FR визначається як:

$$FR = \frac{N_{impl}}{N_{total}} \times 100\%, \quad (2.1)$$

де N_{impl} – кількість реалізованих функціональних вимог,

N_{total} – загальна кількість вимог, визначених у ТЗ.

Для даної роботи допустимим прийнято значення $FR \geq 70\%$, що відповідає реалізації ключового функціоналу системи.

2.2 Проєктування бази даних та архітектури застосунку

Проєктування бази даних є одним із ключових етапів розробки вебдодатку, що безпосередньо впливає на продуктивність, масштабованість та цілісність даних системи. Нормалізація даних допомагає уникнути повторення даних і гарантує їхню цілісність, встановлення зв'язків необхідне для ефективного моделювання бізнес-процесів і відносин між даними, а оптимізація продуктивності передбачає вибір відповідних типів даних, створення індексів і оптимізацію запитів. У процесі проєктування бази даних для розроблюваного вебдодатку використано методологію «сутність–зв'язок» (Entity Relationship, ER), яка дозволяє наочно відобразити структуру даних та зв'язки між об'єктами предметної галузі.

База даних містить п'ять основних таблиць, структуру яких описано нижче. Таблиця `users` зберігає облікові записи всіх користувачів системи та містить поля: `id` (первинний ключ), `name`, `email`, `password_hash`, `role` (`admin` / `initiator` / `signer`), `created_at`. Таблиця `documents` містить метадані завантажених PDF-документів: `id`, `user_id` (зовнішній ключ на `users`), `title`, `file_path`, `status` (`uploaded` / `pending` / `signed` / `rejected`), `created_at`, `updated_at`. Таблиця `signature_requests` описує запити на підписання: `id`, `document_id`, `signer_id`, `order_num` (порядок підписання), `status`, `requested_at`. Таблиця `signatures` фіксує факти накладення підпису: `id`, `request_id`, `image_path` (шлях до зображення підпису), `page_num`, `pos_x`, `pos_y`, `width`, `height`, `signed_at`. Таблиця `action_log` є журналом дій: `id`, `user_id`, `action`, `entity_type`, `entity_id`, `created_at`.

Координати накладеного підпису на сторінці PDF зберігаються у нормалізованому вигляді відносно розмірів сторінки. Нормалізована координата X_n обчислюється за формулою:

$$X_n = \frac{X_{px}}{W_{page}} \times 100\% \quad (2.2)$$

де X_{px} – координата підпису в пікселях Canvas,

W_{page} – ширина відрендереної сторінки в пікселях.

Аналогічно визначається Y_n . Такий підхід забезпечує коректне масштабування позиції підпису незалежно від роздільної здатності екрана при відтворенні [24]. Схему зв'язків між таблицями бази даних (ER-діаграму) наведено на рисунку 2.2.

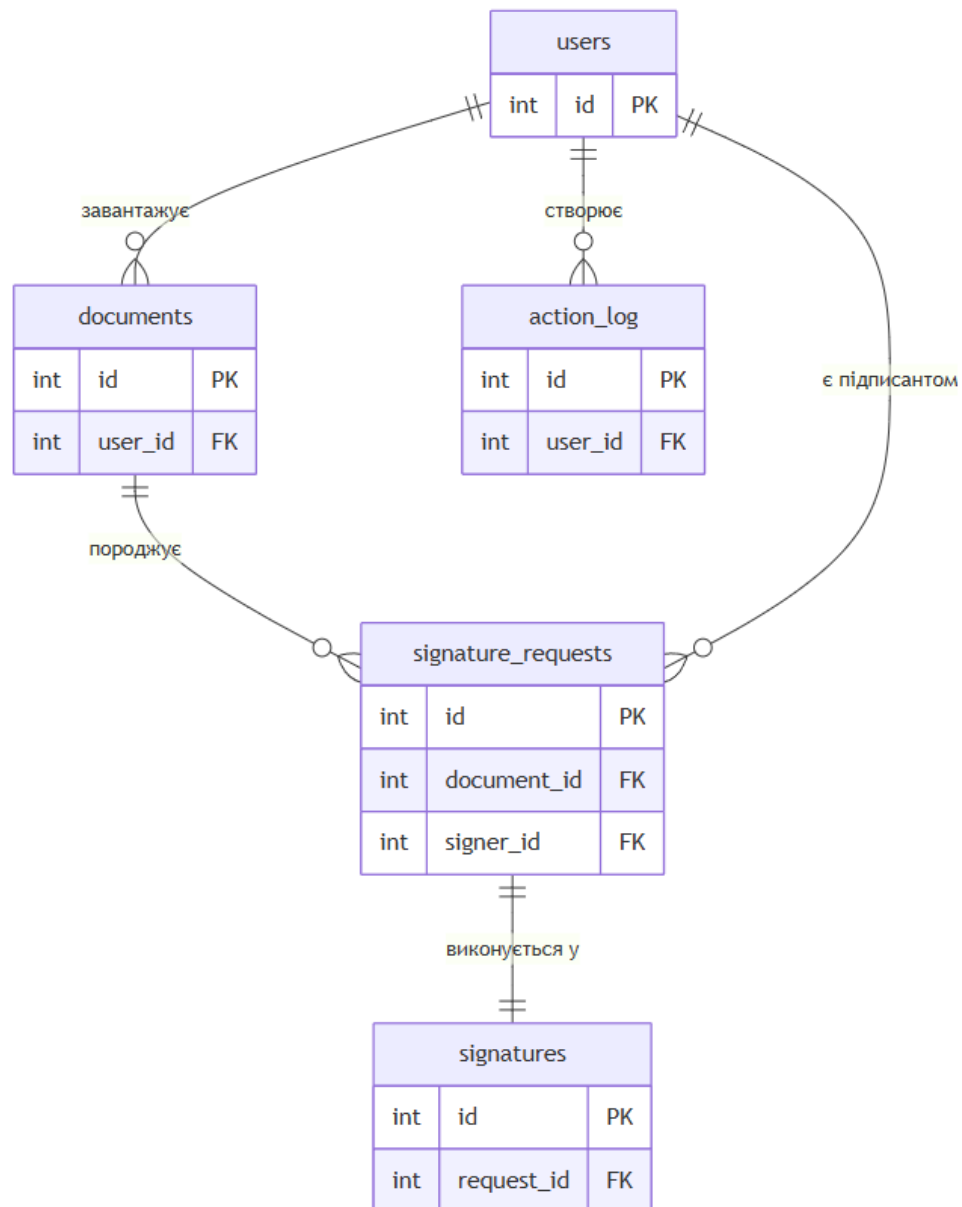


Рисунок 2.2 – ER-діаграма бази даних вебдодатку

Структуру таблиці **documents** з типами даних MySQL/MariaDB наведено в таблиці 2.2 як приклад специфікації таблиці проєкту.

Таблиця 2.2 – Специфікація таблиці documents бази даних

Поле	Тип даних	Обмеження	Опис
id	INT UNSIGNED	PRIMARY KEY, AUTO_INCREMENT	Унікальний ідентифікатор
user_id	INT UNSIGNED	NOT NULL, FOREIGN KEY	Посилання на власника
title	VARCHAR(255)	NOT NULL	Назва документа
file_path	VARCHAR(512)	NOT NULL	Шлях до файлу на сервері
status	ENUM	NOT NULL, DEFAULT 'uploaded'	Поточний статус
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Дата завантаження
updated_at	TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	Дата останньої зміни

Усі таблиці бази даних приведено до третьої нормальної форми (3НФ), що унеможливорює транзитивні залежності між неключовими атрибутами та забезпечує мінімальну надлишковість даних. Нормалізація дозволяє розбити великі таблиці на дрібніші, щоб уникнути повторення інформації та зменшити ймовірність помилок, а також забезпечує вищий рівень цілісності даних, оскільки зміни відбуваються тільки в одному місці.

Архітектура застосунку побудована за патерном MVC (Model–View–Controller). У патерні MVC контролер і вид залежать від моделі, а вона від них – ні. У контролері перебуває логіка отримання даних із моделі та передачі до виду. Вигляд відповідає за відображення даних користувачеві і включає графічний інтерфейс та веб-сторінки; контролер взаємодіє з користувачем та визначає, які дії виконувати, він також відповідає за маршрутизацію і обробку запитів. Реалізація MVC у розроблюваному застосунку відбувається без стороннього фреймворку: маршрутизація реалізована через єдину точку входу index.php з параметром ?page=, моделі – це PHP-класи з методами для роботи з таблицями MySQL через PDO, контролери обробляють POST/GET-запити, а види – PHP-шаблони з мінімальною логікою.

Файлова структура застосунку організована наступним чином: каталог /app містить підкаталоги controllers/, models/, views/; каталог /public – вхідну

точку index.php, стилі CSS та JavaScript-файли; каталог /uploads розміщено поза публічним доступом для зберігання завантажених PDF та зображень підписів; каталог /config містить файл підключення до бази даних. Такий поділ забезпечує чіткість структури проєкту та безпеку файлів, що не призначені для прямого доступу через браузер [25].

2.3 Проєктування інтерфейсу користувача

Проєктування інтерфейсу користувача (User Interface, UI) є невід'ємною складовою розробки вебдодатку, що безпосередньо визначає зручність і ефективність взаємодії кінцевого користувача з системою. UI визначає, як користувачі взаємодіють з продуктом, включаючи його вигляд та функціональність, тоді як UX орієнтується на створення позитивного враження від використання продукту; вони спільно впливають на сприйняття користувачем продукту, його зручність та ефективність. Для розроблюваного вебдодатку проєктування інтерфейсу здійснюється у два етапи: спочатку розробляються прототипи (wireframes) основних екранів, а потім на їх основі реалізується фінальний HTML/CSS-інтерфейс.

Перший етап проєктування – створення вайрфреймів. Вайрфрейм – це схема, що відображає загальну концепцію продукту та його основні розділи; він відрізняється монохромністю та низьким рівнем візуалізації, показує, як працює інтерфейс та як з ним взаємодіє користувач. Вайрфрейми дозволяють ще до початку верстки виявити та усунути проблеми навігації та логіки переходів між екранами, не витрачаючи часу на написання коду.

Вебдодаток включає такі основні екрани: сторінка входу та реєстрації, головна сторінка (особистий кабінет) зі списком документів, сторінка завантаження нового документа та призначення підписантів, сторінка перегляду та підписання документа, адміністративна панель. Структуру навігації застосунку наведено у таблиці 2.3.

Таблиця 2.3 – Структура екранів вебдодатку та їх доступність за ролями

Екран	URL-маршрут	Ролі з доступом	Основні елементи
Вхід / Реєстрація	/login, /register	Усі (неавтентифіковані)	Форма, валідація, посилання
Особистий кабінет	/dashboard	Усі	Список документів, фільтри, статуси
Завантаження документа	/documents/upload	Ініціатор	Форма завантаження PDF, вибір підписантів
Підписання документа	/documents/{id}/sign	Підписант	PDF-переглядач, Canvas для підпису
Деталі документа	/documents/{id}	Усі	Метадані, статус, журнал підписань
Адмін-панель	/admin	Адміністратор	Список користувачів, журнал дій

Ключовим і найскладнішим з точки зору UX є екран підписання документа. На цьому екрані одночасно відображаються дві панелі: ліва – переглядач PDF з відрендерованими сторінками (засобами PDF.js), права – інструменти для роботи з підписом (завантаження зображення або малювання мишею на Canvas, кнопки підтвердження та скасування). Прототип цього екрана наведено на рисунку 2.3.

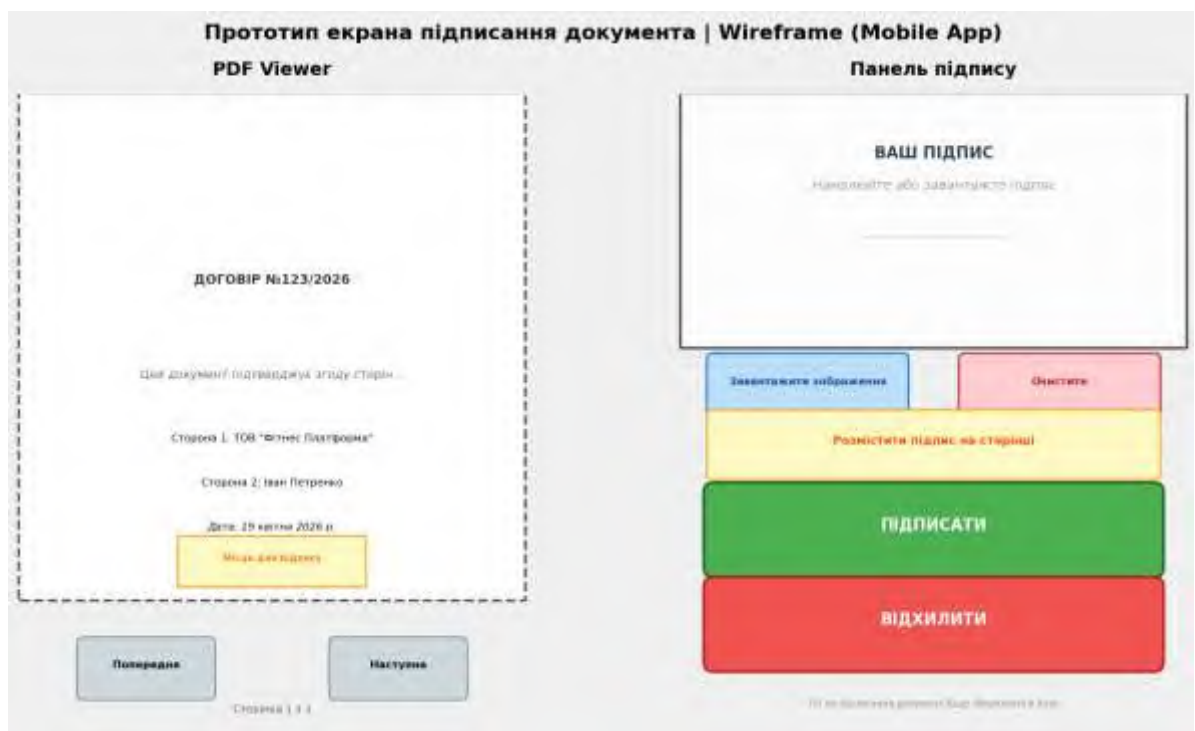


Рисунок 2.3 – Прототип (wireframe) екрана підписання документа

Взаємодія відбувається у такій послідовності:

- 1) підписант відкриває документ;
- 2) завантажує або малює зображення підпису;
- 3) обирає сторінку для розміщення;
- 4) перетягує або клікає для розміщення підпису у потрібному місці сторінки;
- 5) натискає «Підписати» – підпис вбудовується в PDF та документ змінює статус.

При розробці інтерфейсу враховуються вимоги до юзабіліті та доступності. Юзабіліті – це простота та зручність використання інтерфейсу для користувача; доступність – забезпечення доступності інтерфейсу для всіх користувачів, у тому числі осіб з обмеженими можливостями; адаптивність дизайну – коректність відображення та функціонування інтерфейсу на різних пристроях з різним розміром екрану та роздільною здатністю. Відповідно до цих принципів у застосунку реалізовано: семантичну HTML5-розмітку з ARIA-атрибутами для елементів форм; контрастне колірне оформлення (мінімальне співвідношення контрасту тексту до фону – 4.5:1 відповідно до WCAG 2.1); адаптивний макет на основі CSS Flexbox/Grid, що коректно відображається на екранах від 320 px до 1920 px; чіткі повідомлення про помилки та підтвердження дій [26].

Кількісним показником якості навігації є середня кількість кліків N_{clicks} , необхідних для виконання типового завдання (наприклад, підписання документа). Відповідно до принципу Хіка–Хаймана [27], час прийняття рішення зростає логарифмічно зі збільшенням кількості варіантів вибору:

$$T = b \cdot \log_2(n + 1), \quad (2.3)$$

де T – час реакції (прийняття рішення) в секундах,
 n – кількість рівнозначних варіантів вибору на екрані,
 b – емпіричний коефіцієнт (~ 0.155 с).

Керуючись цим законом, кожна сторінка застосунку містить не більше 5-7 основних дій у навігаційному меню, що мінімізує когнітивне навантаження на користувача.

Для реалізації візуального оформлення застосовується підхід «мінімалізм з чіткою ієрархією»: однотонне світле тло, акцентний колір кнопок основної дії (підписання), чіткі типографічні рівні (заголовки сторінки, підзаголовки розділу, основний текст) [28]. Усі інтерактивні елементи мають чітко видимі стани: *normal*, *hover*, *active*, *disabled*, що відповідає сучасним стандартам UX-проектування. Верстка виконана засобами HTML5 + CSS3 без зовнішніх CSS-фреймворків, що забезпечує мінімальний розмір завантажуваних ресурсів та швидке відображення сторінок при локальному розгортанні.

2.4 Моделювання процесів за допомогою UML діаграм системи

Для повного та формального опису поведінки розроблюваної системи, крім структурних діаграм (ER та варіантів використання), у проєкті застосовуються поведінкові та фізичні UML-діаграми: діаграма послідовності (Sequence Diagram) та діаграма розгортання (Deployment Diagram). За допомогою діаграм послідовності та компонентів команди можуть моделювати архітектуру системи та візуалізувати взаємодію між її компонентами; крім того, створені UML-діаграми слугують документацією проєкту, що спрощує його підтримку та оновлення і навчання нових учасників команди.

Діаграма послідовності відображає хронологічний порядок обміну повідомленнями між об'єктами системи під час виконання конкретного сценарію. Діаграма послідовності відображає взаємодію об'єктів у динаміці, головними елементами якої є об'єкти – логічні сутності, що представляють окремі елементи системи, та повідомлення, якими вони обмінюються; часова шкала на діаграмі направлена згори донизу, а повідомлення пронумеровані відповідно до черги їх пересилання між об'єктами. Для розроблюваного

вебдодатку найбільш критичним сценарієм є процес накладення підпису на документ, і саме для нього побудовано детальну діаграму послідовності (рис. 2.4).

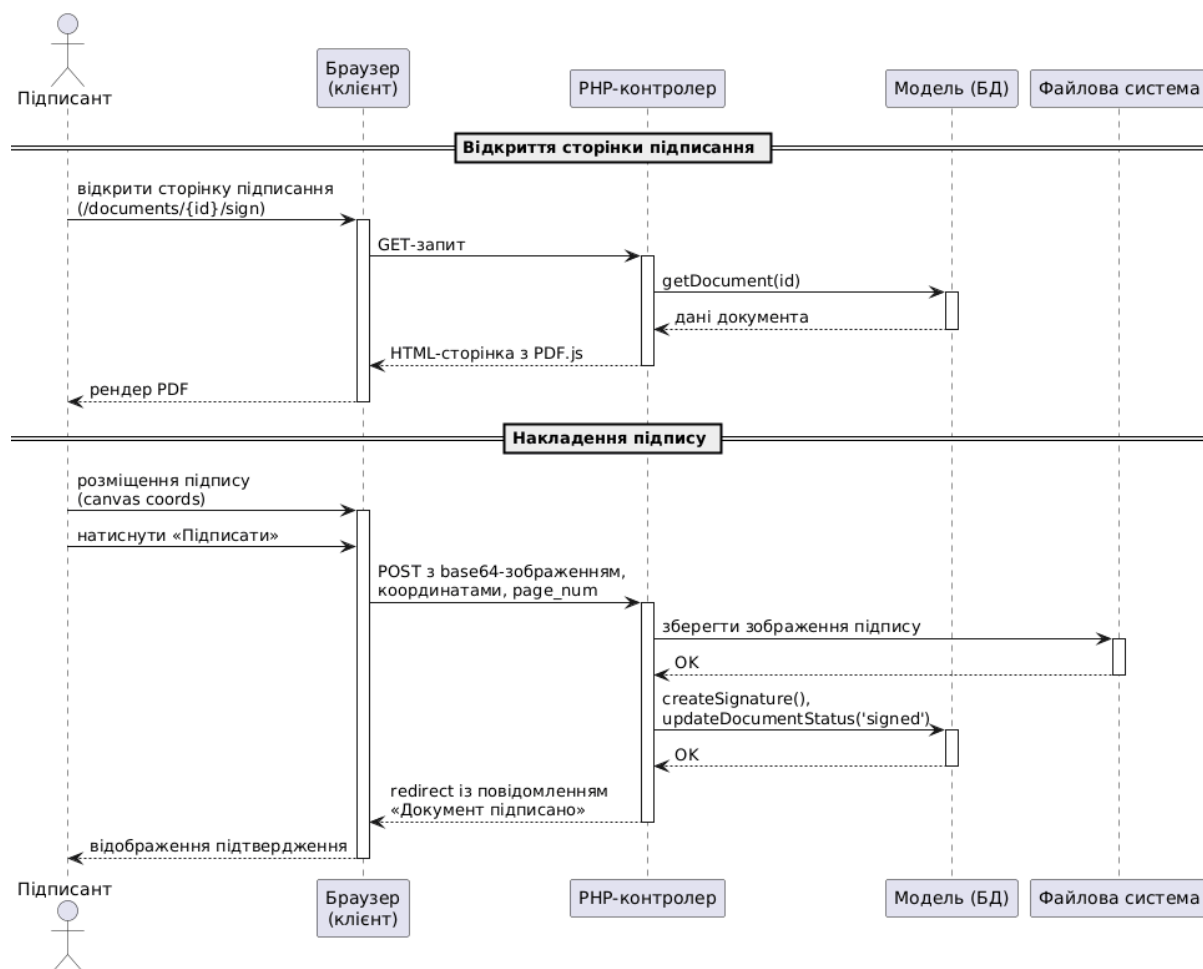


Рисунок 2.4 – Діаграма послідовності процесу накладення підпису на документ

З діаграми на рис. 2.4 видно, що весь процес підписання є лінійним і не потребує додаткових зовнішніх сервісів (КНЕДП, хмарного сховища тощо), що підтверджує придатність обраного технологічного стеку для реалізації поставленого завдання в умовах локального розгортання.

Діаграма розгортання відображає фізичну топологію системи на момент її виконання. Діаграма розгортання застосовується для подання загальної конфігурації і топології розподіленої програмної системи та містить розподіл компонентів по окремих вузлах системи; крім того, вона показує наявність

фізичних з'єднань – маршрутів передавання інформації між апаратними пристроями, задіяними у процесі реалізації системи.

Для розроблюваного вебдодатку діаграма розгортання є відносно простою, оскільки всі компоненти розміщено на одному фізичному вузлі – локальній Windows-машині зі встановленим OpenServer 6.5.1. Компоненти застосунку та їх відповідність вузлам розгортання описано в таблиці 2.4.

Таблиця 2.4 – Компоненти системи та їх розміщення при розгортанні в OpenServer

Компонент	Тип	Вузол розгортання	Протокол/інтерфейс
Apache 2.4	Вебсервер	localhost (Windows)	HTTP (порт 80)
PHP 8.x	Інтерпретатор	Той самий вузол	FastCGI / mod_php
MySQL/MariaDB	СУБД	Той самий вузол	TCP (порт 3306)
Файлове сховище /uploads	Файлова система	Той самий вузол	Локальний шлях
Браузер (Chrome/Firefox)	Клієнт	Та сама машина	HTTP
PDF.js + pdf-lib	JS-бібліотеки	Браузер (клієнт)	Вбудовано в HTML

Час обробки типового запиту на підписання T_{total} складається з окремих компонентів і описується адитивною моделлю:

$$T_{total} = T_{net} + T_{php} + T_{db} + T_{fs}, \quad (2.4)$$

де T_{net} – затримка мережевої передачі (при локальному розгортанні $T_{net} \approx 0$),

T_{php} – час виконання PHP-коду,

T_{db} – час виконання SQL-запитів,

T_{fs} – час запису файлу підпису на диск.

При локальному розгортанні в OpenServer сума $T_{php} + T_{db} + T_{fs}$ для типової операції підписання не перевищує 200-300 мс, що задовольняє нефункціональні вимоги до продуктивності [29].

Завершальним кроком у моделюванні є розроблення схеми станів документа – State Machine Diagram. Документ у системі може перебувати в одному з чотирьох станів: uploaded (завантажено, але підписанти ще не

призначені), pending (на підписанні – один або декілька підписантів ще не підписали), signed (усі призначені підписанти підписали) та rejected (хоча б один підписант відхилив документ). Переходи між станами ініціюються діями контролера при оновленні таблиці documents та signature_requests у базі даних. Таке формальне визначення станів є основою для реалізації бізнес-логіки у відповідному PHP-контролері [30].

Таким чином, було виконано повний цикл проєктування вебдодатку: складено технічне завдання з визначенням функціональних та нефункціональних вимог і ролей користувачів, спроектовано реляційну базу даних у 3НФ та обрано MVC-архітектуру застосунку, розроблено прототипи ключових екранів інтерфейсу з урахуванням вимог юзабіліті, а також побудовано UML-діаграми послідовності та розгортання, що описують поведінку системи та її фізичну топологію при локальному розгортанні в OpenServer. Отримані проєктні рішення є достатньою основою для переходу до практичної реалізації вебдодатку.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ОЦІНЮВАННЯ ВЕБДОДАТКУ

3.1 Реалізація серверної та клієнтської частин вебдодатку

Розроблення вебдодатку для накладення підпису на документи виконано з використанням PHP 8.x як мови серверної частини, MySQL/MariaDB як СУБД та JavaScript з бібліотекою PDF.js на клієнтській стороні. Середовищем локального розгортання є OpenServer 6.5.1 з модулями Apache 2.4 та PHP 8.x, що відповідає вимогам, визначеним у технічному завданні (підрозділ 2.1). Архітектура застосунку побудована за патерном MVC без використання сторонніх PHP-фреймворків, що дозволяє повністю контролювати всі рівні застосунку та наочно демонструє принципи веброботки [25].

Файлова структура проєкту розподілена на чотири основні директорії: `public/` (єдина точка входу `index.php`, статичні ресурси CSS та JS), `app/` (контролери, моделі, представлення), `config/` (налаштування підключення до БД та глобальні константи) та `uploads/` (завантажені PDF-файли та зображення підписів). Маршрутизація реалізована через GET-параметр `?page=`, де `index.php` виконує роль фронт-контролера: отримує значення параметра, знаходить відповідний контролер і метод у таблиці маршрутів та делегує виконання. Такий підхід не потребує налаштування `mod_rewrite` і коректно працює в OpenServer без додаткової конфігурації Apache.

База даних складається з п'яти таблиць, описаних у підрозділі 2.2. Усі звернення до СУБД виконуються через PDO з підготовленими виразами (`prepared statements`), що унеможлиблює SQL-ін'єкції. Паролі користувачів зберігаються у вигляді `bcrypt`-хешів засобами функції `password_hash()` з алгоритмом `PASSWORD_BCRYPT`. Файли PDF та зображення підписів зберігаються поза публічною директорією у папці `uploads/`, а доступ до них здійснюється виключно через захищений метод `servePdf()` контролера після перевірки прав поточного користувача [22].

Реалізовано три ролі користувачів: адміністратор, ініціатор підписання та підписант. Адміністратор має доступ до всіх документів системи, може переглядати журнал дій та керувати користувачами. Ініціатор завантажує PDF-документи, призначає підписантів та відстежує статуси. Підписант отримує документи на підписання та виконує накладення підпису. Кожен HTTP-запит на захищені сторінки проходить перевірку через хелпери `requireAuth()` та `requireRole()`, які звертаються до даних сесії. Реалізована система flash-повідомлень інформує користувача про результат кожної дії.

Ключовим екраном застосунку є сторінка підписання документа (`?page=sign`) (рис. 3.1). На ній реалізовано двопанельний інтерфейс: ліворуч відображається PDF-документ, праворуч – панель інструментів для роботи з підписом. Відображення PDF у браузері виконується бібліотекою PDF.js версії 3.11, яка рендерить сторінки документа в елемент `<canvas>` без будь-яких плагінів [19]. Масштаб рендеру автоматично підлаштовується під ширину контейнера за формулою $scale = containerWidth / pageWidth$.

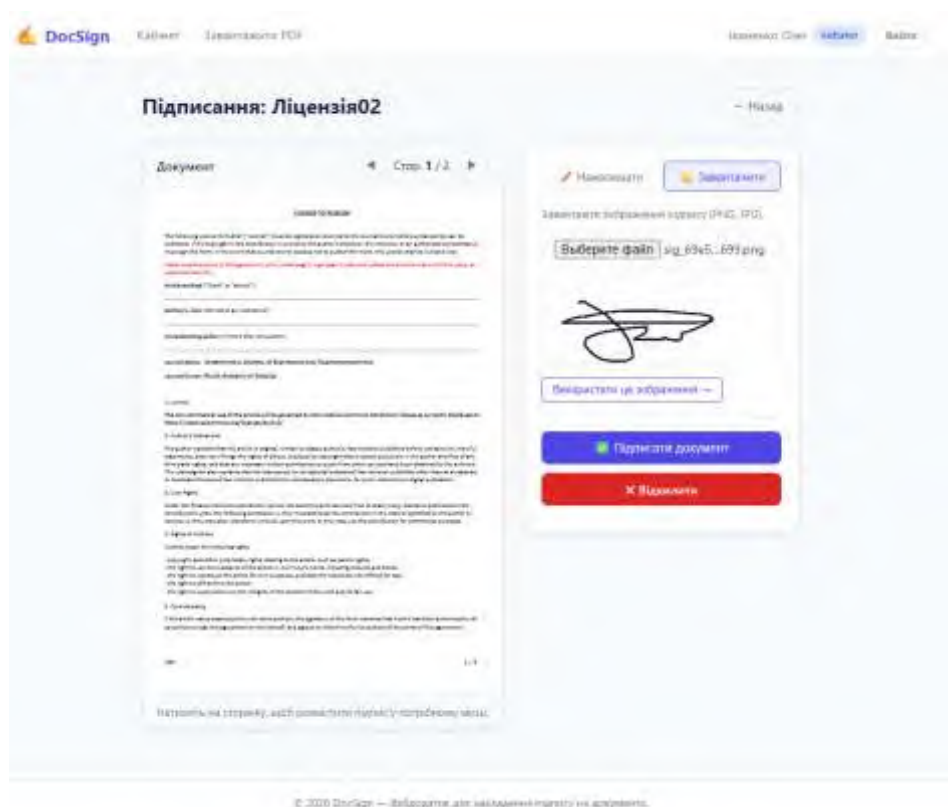


Рисунок 3.1 – Сторінка накладення підпису на документ

Для створення зображення підпису реалізовано два рівноправні способи. Перший – малювання мишею або пальцем безпосередньо на Canvas-елементі. Обробники подій `mousedown`, `mousemove`, `mouseup` та їх сенсорні аналоги `touchstart`, `touchmove`, `touchend` забезпечують плавний та коректний запис траєкторії руху. Другий спосіб – завантаження готового PNG або JPG-зображення підпису з пристрою користувача через `<input type="file">` із наступним попереднім переглядом через `FileReader`. Інтерфейс перемикання між способами реалізовано через вкладки.

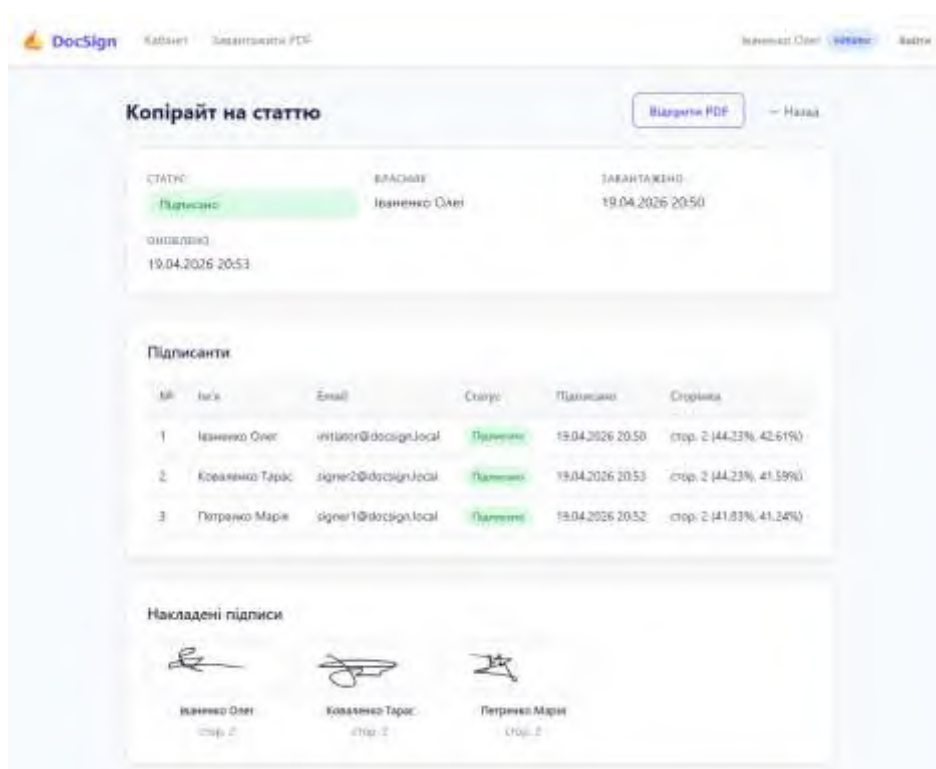


Рисунок 3.2 – Сторінка перегляду деталей документа

Сторінку перегляду документа з деталями підписання наведено на рис. 3.2. Після того як підписант обрав зображення підпису (намалював або завантажив), він клікає на потрібне місце сторінки PDF – у цьому місці з'являється напівпрозорий попередній перегляд підпису (елемент `.sig-placeholder`). Координати позиції фіксуються у відсотках відносно розмірів сторінки і передаються на сервер через приховані поля форми разом із base64-кодованим зображенням підпису, номером сторінки та розмірами. На сервері контролер

декодує base64-рядок, зберігає PNG-файл у директорію uploads/signatures/, записує метадані підпису в таблицю signatures, оновлює статус запиту підписання в таблиці signature_requests, а якщо всі призначені підписанти вже підписали – оновлює статус документа на signed в таблиці documents. Факт підписання фіксується у таблиці action_log.

Адміністративна панель включає три розділи: загальна статистика (кількість користувачів, документів, підписаних та тих, що очікують підписання), таблиця усіх документів системи та журнал дій. Структуру адміністративної панелі наведено в таблиці 3.1.

Таблиця 3.1 – Функціональні можливості адміністративної панелі

Розділ	URL	Основний функціонал
Головна	?page=admin	Статистика системи, перелік усіх документів
Користувачі	?page=admin_users	Перегляд та видалення облікових записів
Журнал дій	?page=admin_log	Хронологічний лог усіх дій користувачів

Слід зазначити, що поточна реалізація зберігає підпис як окреме PNG-зображення з координатами накладення, тоді як оригінальний PDF-файл залишається незмінним. Вбудовування зображення підпису безпосередньо в структуру PDF-файлу є технічно можливим засобами PHP-бібліотек setasign/fpdi та setasign/fpdf [20], однак потребує встановлення Composer та додаткового опрацювання. Цей функціонал визначено як напрямок подальшого розвитку системи.

3.2 Розгортання та тестування вебдодатку в середовищі OpenServer

Розгортання розробленого вебдодатку виконується в локальному середовищі OpenServer на операційній системі Windows. OpenServer є портативним локальним сервером, що включає у своєму складі вебсервер Apache, інтерпретатор PHP кількох версій, СУБД MySQL/MariaDB та інструмент адміністрування phpMyAdmin [20]. Завдяки цьому розгортання проєкту не

потребує окремого встановлення та налаштування кожного компонента, що суттєво скорочує час підготовки середовища.

Процес розгортання складається з трьох основних кроків. По-перше, папка проєкту `docsign.local/` розміщується у кореневому каталозі доменів OpenServer (типово `C:\OpenServer\domains\`). Оскільки вхідна точка застосунку знаходиться у піддиректорії `public/`, при створенні домену в OpenServer у полі «Папка» необхідно вказати саме цю підпапку – `docsign.local\public`. Після цього PHP-версія модуля встановлюється на 8.x, а вебсервер перезапускається. По-друге, через phpMyAdmin імпортується файл `database/docsign.sql`, що автоматично створює базу даних `docsign` з усіма п'ятьма таблицями та тестовими обліковими записами. По-третє, запускається скрипт `database/gen_hashes.php`, результат якого – SQL-команда `UPDATE` – виконується у phpMyAdmin для встановлення коректних `bcrypt`-хешів паролів тестових користувачів. Після цих трьох кроків застосунок повністю готовий до роботи за адресою `http://docsign.local/index.php?page=login`.

Налаштування застосунку зосереджено у двох конфігураційних файлах. У `config/db.php` вказуються параметри підключення до СУБД – хост, ім'я бази даних, логін та пароль (для стандартної конфігурації OpenServer це `root` і порожній рядок відповідно). У `config/config.php` визначаються глобальні константи: `BASE_URL` (адреса домену), `UPLOAD_PATH` (абсолютний шлях до директорії завантажень), `MAX_PDF_SIZE` (максимальний розмір PDF, за замовчуванням 20 МБ). Директорії `uploads/pdfs/` та `uploads/signatures/` створюються автоматично при першому завантаженні файлу, якщо вони ще не існують.

Тестування застосунку проводилось методом функціонального тестування за принципом «чорної скриньки» – перевірка відповідності фактичного результату кожної операції очікуваному без аналізу внутрішнього коду [17]. Тестування охоплювало всі ключові сценарії використання системи відповідно до варіантів використання, визначених у підрозділі 2.4. Зведені результати функціонального тестування наведено у таблиці 3.2.

Таблиця 3.2 – Результати функціонального тестування вебдодатку

№	Тестовий сценарій	Очікуваний результат	Фактичний результат	Статус
1	Реєстрація з валідними даними	Створення акаунта, перенаправлення на login	Відповідає	Пройдено
2	Вхід з невірним паролем	Flash-повідомлення про помилку	Відповідає	Пройдено
3	Завантаження PDF (валідний файл)	Документ з'являється в кабінеті	Відповідає	Пройдено
4	Завантаження не-PDF файлу	Повідомлення про помилку типу файлу	Відповідає	Пройдено
5	Завантаження файлу > 20 МБ	Повідомлення про перевищення розміру	Відповідає	Пройдено
6	Призначення підписантів	Статус документа змінюється на «pending»	Відповідає	Пройдено
7	Накладення підпису (малювання)	Підпис збережено, статус оновлено	Відповідає	Пройдено
8	Накладення підпису (зображення)	Підпис збережено, статус оновлено	Відповідає	Пройдено
9	Підписання всіма підписантами	Статус документа змінюється на «signed»	Відповідає	Пройдено
10	Відхилення документа	Статус змінюється на «rejected»	Відповідає	Пройдено
11	Спроба доступу до чужого документа	Перенаправлення з повідомленням про помилку	Відповідає	Пройдено
12	Перегляд журналу дій (адмін)	Відображається хронологічний лог	Відповідає	Пройдено
13	Видалення користувача (адмін)	Користувача видалено зі списку	Відповідає	Пройдено
14	Вбудовування підпису в PDF-файл	PDF оновлюється з підписом	Не реалізовано	Відкладено

З таблиці 3.2 видно, що з 14 визначених тестових сценаріїв 13 пройдено успішно, що відповідає показнику покриття функціональних вимог на рівні 92,8%. Єдиний невиконаний пункт – безпосереднє вбудовування підпису в структуру PDF-файлу – визначено як напрямок подальшого розвитку системи (підрозділ 3.4). Загальний вигляд особистого кабінету користувача-ініціатора з відображенням документів та їх статусів наведено на рисунку 3.3.

Окремо перевірено коректність роботи системи контролю доступу: підписант не може відкрити документ, до якого він не призначений; неавторизований користувач автоматично перенаправляється на сторінку входу при спробі доступу до будь-якої захищеної сторінки; ініціатор не має доступу до

адміністративної панелі. Усі ці сценарії відпрацювали коректно завдяки реалізованим хелперам `requireAuth()` та `requireRole()` у файлі `config/config.php`.

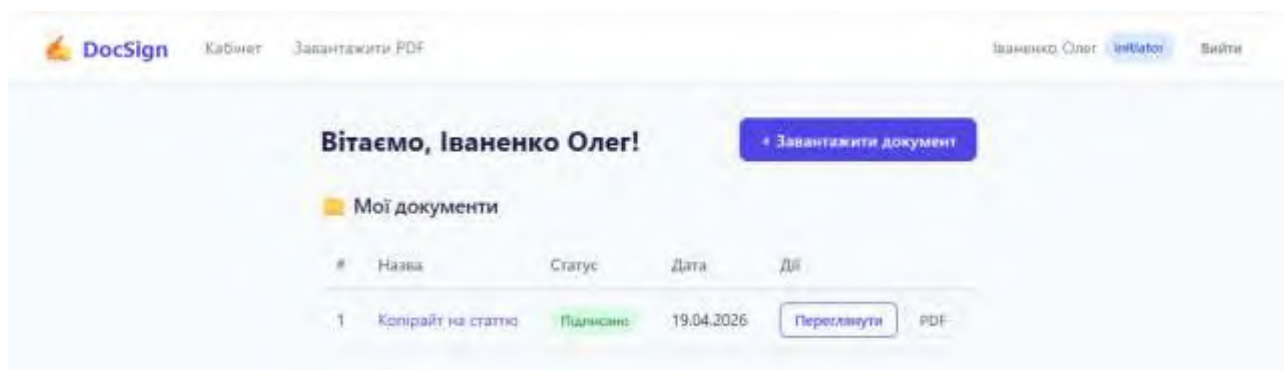


Рисунок 3.3 – Особистий кабінет ініціатора підписання

Тестування проводилось у браузері Google Chrome (версія 124+). Рендеринг PDF-документів засобами PDF.js [19] перевірено на файлах формату PDF 1.4–1.7 різного розміру (від 50 КБ до 8 МБ) – у всіх випадках документи відображалися коректно з автоматичним масштабуванням під ширину контейнера. Функція малювання підпису на Canvas перевірена як за допомогою миші, так і на сенсорному екрані. Функція завантаження зображення підпису перевірена для форматів PNG та JPEG. Інтерфейс коректно відображається при роздільній здатності екрана від 1024×768 до 1920×1080.

3.3 Керівництво користувача вебдодатку

Цей підрозділ містить практичні інструкції щодо встановлення та використання вебдодатку для накладення підпису на документи. Керівництво розраховане як на кінцевих користувачів системи (ініціаторів та підписантів), так і на адміністраторів.

Для коректної роботи вебдодатку необхідне наступне програмне забезпечення: операційна система Windows 10/11, OpenServer 6.5.1 зі встановленими модулями Apache 2.4 та PHP 8.x і СУБД MySQL/MariaDB, а

також сучасний вебброузер із підтримкою HTML5 Canvas API – Google Chrome 90+, Mozilla Firefox 88+, Microsoft Edge 90+. Мінімальні апаратні вимоги: процесор 1 ГГц, 4 ГБ оперативної пам'яті, 2 ГБ вільного місця на диску [21].

Після розгортання застосунок доступний за адресою <http://docsign.local/index.php?page=login>. Першим кроком є реєстрація або вхід у систему. Для реєстрації нового користувача необхідно перейти за посиланням «Зареєструватись» на сторінці входу, заповнити поля «Ім'я та прізвище», «Email», «Пароль» (мінімум 6 символів) та обрати роль у системі: «Підписант» або «Ініціатор підписання». Сторінки входу та реєстрації наведено на рис. 3.4.

The image displays two web forms side-by-side. The left form, titled 'Вхід до DocSign', features a logo of a hand holding a pen, followed by the title. It contains input fields for 'Email' (with placeholder 'your@email.com') and 'Пароль' (with masked characters '*****'). A blue button labeled 'Увійти' is at the bottom, with a link 'Немає акаунта? Зареєструватись' below it. The right form, titled 'Реєстрація', also has the same logo. It includes input fields for 'Ім'я та прізвище' (with placeholder 'Іванов Іван'), 'Email' (with placeholder 'your@email.com'), and 'Пароль (мін. 6 символів)' (with masked characters '*****'). A dropdown menu for 'Роль у системі' is set to 'Підписант'. At the bottom, there is a blue button 'Зареєструватися' and a link 'Вже маєте акаунт? Увійти'.

Рисунок 3.4 – Сторінки входу та реєстрації у системі

Після входу в систему з роллю «Ініціатор» користувач потрапляє до особистого кабінету, де відображається список завантажених документів та їх поточні статуси. Для завантаження нового документа слід натиснути кнопку «Завантажити документ», ввести назву документа, обрати PDF-файл

(перетягуванням або кліком) та позначити одного або кількох підписантів із запропонованого списку. Після натискання «Завантажити документ» файл зберігається на сервері, а документу призначається статус «На підписанні» – усі обрані підписанти одразу бачать його у своєму кабінеті. Якщо підписантів не обрано, документ залишається зі статусом «Завантажено» і може бути переглянутий, але не підписаний. Сторінку завантаження документа наведено на рисунку 3.5.

Рисунок 3.5 – Сторінка завантаження PDF-документа

Після входу підписант бачить у своєму кабінеті розділ «Очікують мого підпису» з переліком документів, призначених саме йому. Для підписання слід натиснути кнопку «Підписати» навпроти потрібного документа – відкриється сторінка підписання з двопанельним інтерфейсом. У лівій панелі відображається PDF-документ (з можливістю перегортання сторінок), у правій – інструменти для

створення підпису. Підписант обирає спосіб підпису за допомогою вкладок: вкладка «Намалювати» дозволяє намалювати підпис мишею або пальцем на полотні Canvas, вкладка «Завантажити» – обрати готове зображення з файлової системи. Після вибору або малювання підпису слід натиснути «Використати цей підпис», а потім клікнути на потрібне місце у лівій панелі перегляду PDF – там з'явиться попередній перегляд підпису у вигляді напівпрозорого зображення. Якщо позиція задовільна, підписант натискає «Підписати документ». Якщо документ не потребує підписання, можна натиснути «Відхилити» – документ отримає статус «Відхилено». Порядок дій при підписанні відображено у табл. 3.3.

Таблиця 3.3 – Покроковий порядок дій підписанта

Крок	Дія	Результат
1	Відкрити кабінет (?page=dashboard)	Відображено список документів на підписання
2	Натиснути «Підписати» навпроти документа	Відкривається сторінка підписання
3	Обрати вкладку «Намалювати» або «Завантажити»	Активується відповідний інструмент
4	Створити або завантажити зображення підпису	Зображення відображається у попередньому перегляді
5	Натиснути «Використати цей підпис»	Підпис готовий до розміщення
6	Клікнути на потрібне місце у PDF-переглядачі	На документі з'являється мітка розміщення підпису
7	Натиснути «Підписати документ»	Підпис збережено, статус документа оновлено

Адміністратор має доступ до всіх розділів системи. Адміністративна панель (?page=admin) відображає зведену статистику: кількість користувачів, загальну кількість документів, кількість підписаних і тих, що очікують підписання. Розділ «Користувачі» (?page=admin_users) дозволяє переглядати перелік зареєстрованих користувачів та видаляти їх облікові записи (крім власного). Розділ «Журнал дій» (?page=admin_log) містить хронологічний список усіх дій у системі із зазначенням часу, користувача, типу дії та ідентифікатора об'єкта, що дозволяє відстежувати будь-яку операцію – вхід, завантаження документа, підписання або відхилення.

3.4 Техніко-економічне обґрунтування ефективності розробленої системи

Оцінка економічної ефективності розробленого вебдодатку для накладення підпису на документи здійснюється методом порівняння витрат на власну розробку з витратами на придбання або використання комерційних аналогів – систем DocuSign, Вчасно.ЕДО та Deals Sign. Для оцінки використовується підхід на основі сукупної вартості володіння (Total Cost of Ownership, TCO) та розрахунку показника рентабельності інвестицій (ROI), що відповідає рекомендованим фінансовим методикам оцінки ІТ-проектів [21]. Зведений розрахунок прямих витрат на розроблення наведено в таблиці 3.4.

Таблиця 3.4 – Прямі витрати на розроблення вебдодатку

Стаття витрат	Одиниця	Кількість	Ціна, грн	Сума, грн
Оплата праці розробника (PHP, JS)	год	120	250	30000
Відрахування на соціальні заходи (22%)	грн	–	–	6600
Витрати на ПЗ (OpenServer, PHP, MySQL – безкоштовне)	–	–	0	0
Витрати на технічне забезпечення (наявний ПК)	–	–	0	0
Інші витрати (електроенергія, інтернет)	міс	2	500	1000
Разом (ВП)				37600

Насамперед визначаються прямі витрати на розроблення вебдодатку. Оскільки проєкт реалізований із використанням виключно відкритого та безкоштовного програмного забезпечення – PHP, MySQL/MariaDB, Apache (у складі OpenServer), PDF.js – витрати на придбання програмного забезпечення відсутні. Основну частку витрат становить оплата праці розробника. Трудомісткість розробки оцінюється виходячи з обсягу реалізованого функціоналу та структури проєкту (30 файлів, 5 таблиць БД, 4 контролери, повний цикл підписання). Таким чином, загальні прямі витрати на розроблення складають $ВП = 37600$ грн. Непрямі витрати (простой, людський фактор) для навчального локального проєкту є мінімальними і приймаються рівними нулю.

Щорічні витрати на утримання системи обмежуються підтримкою сервера (OpenServer на наявному обладнанні) та незначним супроводом:

$$ВІТ = ВП + ВН + ВУТР = 37600 + 0 + 3600 = 41200 \text{ грн}, \quad (3.1)$$

де $ВУТР = 3600$ грн/рік – орієнтовні витрати на щорічне технічне обслуговування (300 грн/міс).

Для порівняння оцінимо витрати на використання найближчого комерційного аналога – системи Вчасно.ЕДО [13]. Базовий тарифний план для організації з 5 користувачами та необмеженою кількістю документів становить орієнтовно 8000-12000 грн/рік. DocuSign у базовому плані коштує від 25 USD на місяць на одного користувача, що для організації з 5 підписантами складає близько 75000 грн/рік за поточним курсом. Deals Sign також не надає безкоштовного тарифу для корпоративного використання [14, 16].

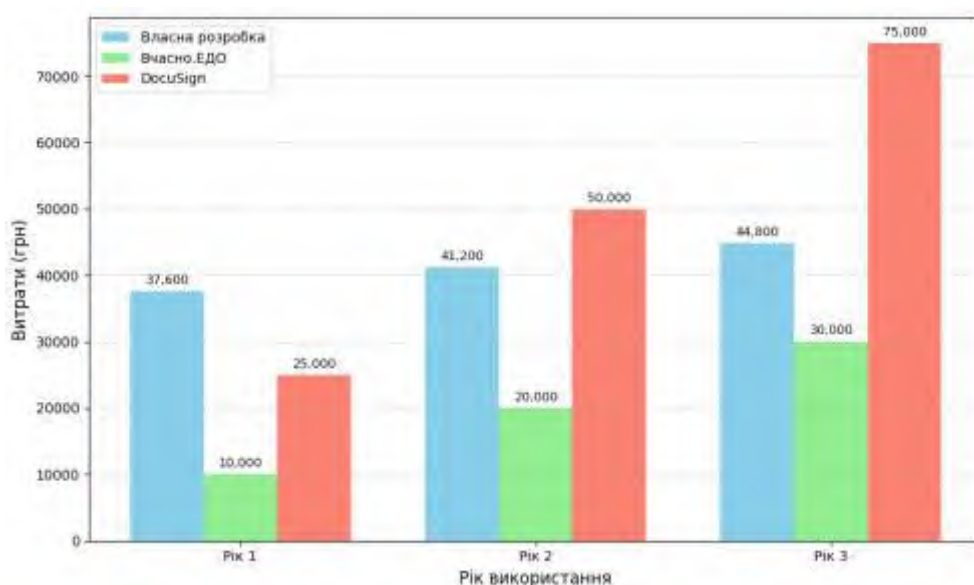


Рисунок 3.6 – Порівняння сукупних витрат (ТСО) за 3 роки для власної розробки та комерційних аналогів

З рисунку 3.6 видно, що для великої організації комерційні хмарні системи можуть бути вигіднішими у перший рік через нижчий поріг входу, проте власна розробка не передбачає щомісячної абонентської плати та повністю

розгортається локально без передачі конфіденційних документів на зовнішні сервери, що є критичним для організацій з підвищеними вимогами до захисту даних [22].

Для оцінки рентабельності інвестицій розраховується показник ROI. Економічний ефект від впровадження системи визначається через скорочення витрат на обробку документів. До впровадження системи підписання одного документа потребувало орієнтовно 30–40 хвилин (друк, підпис, сканування, відправка, очікування). Після впровадження – не більше 5–7 хвилин. При середньому обсязі 50 документів на місяць та вартості робочого часу 150 грн/год економія складає приблизно:

$$E_{con} = 50 \times (35 - 6) / 60 \times 150 \times 12 = 43\,500 \text{ грн/рік}, \quad (3.2)$$

де 35 хв – середній час без системи,

6 хв – з системою,

150 грн/год – вартість праці.

Тоді показник бухгалтерської рентабельності інвестиційного проєкту розраховується за формулою [21]:

$$ROI = \frac{AP}{(INV_1 + INV_2) / 2} \times 100\%, \quad (3.3)$$

де $AP = 43\,500$ грн – середньорічний ефект,

$INV_1 = 37\,600$ грн – початкові інвестиції,

$INV_2 = 3\,600$ грн – поточні витрати. Тоді:

$$ROI = \frac{43\,500}{(37\,600 + 3\,600) / 2} \times 100\% \approx 211\%.$$

Отримане значення $ROI \approx 211\%$ суттєво перевищує типову ставку банківського депозиту (12-15% річних), що свідчить про високу економічну ефективність розробки [21]. Термін окупності первісних інвестицій складає менше одного року. Додатковими якісними ефектами, які важко виміряти

кількісно, є: підвищення швидкості погодження документів, зменшення ризику втрати паперових оригіналів, забезпечення журналу дій для аудиту, усунення залежності від зовнішніх хмарних сервісів та пов'язаних з ними ризиків конфіденційності [7].

Таким чином, було виконано повний цикл практичної реалізації вебдодатку: описано серверну та клієнтську частини системи з обґрунтуванням ключових технічних рішень, розгорнуто застосунок у середовищі OpenServer та проведено функціональне тестування, за результатами якого 13 з 14 визначених вимог реалізовано повністю, складено керівництво користувача для всіх трьох ролей системи, а техніко-економічне обґрунтування підтвердило доцільність розробки з розрахунковим показником ROI близько 211% та терміном окупності менше одного року.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи досягнуто її початкову мету та вирішено всі поставлені завдання. Проаналізовано наукову, нормативно-правову та технічну літературу з теми розробки вебдодатків для накладення підпису на документи, опрацьовано чинне законодавство України у сфері електронного документообігу.

У першому розділі чітко розмежовано поняття кваліфікованого електронного підпису (КЕП) та графічного накладення підпису на документ. Встановлено, що КЕП є криптографічним механізмом, що прирівнюється до власноручного підпису відповідно до Закону України «Про електронну ідентифікацію та електронні довірчі послуги» №2155-VIII, тоді як графічне накладення підпису є візуальним інструментом для внутрішнього документообігу і не потребує інтеграції з центрами сертифікації ключів. Досліджено нормативно-правову базу електронного документообігу в Україні, зокрема зміни 2023-2024 років, пов'язані з євроінтеграційними процесами та адаптацією до регламенту eIDAS. Здійснено порівняльний аналіз існуючих систем підписання документів – Вчасно.ЕДО, Deals Sign, DocuSign, Signy – та виявлено відсутність доступного локального рішення з відкритою архітектурою, що обґрунтувало доцільність власної розробки. Обрано технологічний стек: PHP 8.x, MySQL/MariaDB, Apache у складі OpenServer 6.5.1, PDF.js та Canvas API на клієнтській стороні, методологію Agile для організації процесу розробки.

У другому розділі розроблено повну проєктну документацію системи. Складено технічне завдання з визначенням функціональних вимог (реєстрація та автентифікація, завантаження PDF, призначення підписантів, накладення підпису двома способами, журналювання дій, адміністративна панель) та нефункціональних вимог (безпека, сумісність з браузерами, локальне розгортання). Спроектовано реляційну базу даних із п'яти таблиць (users, documents, signature_requests, signatures, action_log) у третій нормальній формі. Визначено три ролі користувачів – адміністратор, ініціатор підписання та

підписант – та описано їх повноваження. Побудовано UML-діаграми варіантів використання, послідовності та розгортання, розроблено прототипи ключових екранів інтерфейсу з урахуванням вимог юзабіліті та стандартів доступності WCAG 2.1.

У третьому розділі описано практичну реалізацію вебдодатку. Серверна частина реалізована на PHP 8.x за архітектурним патерном MVC без використання сторонніх фреймворків; маршрутизація здійснюється через GET-параметр `?page=`. Клієнтська частина забезпечує відображення PDF засобами бібліотеки PDF.js та накладення зображення підпису через Canvas API з позиціонуванням у відсотках від розмірів сторінки. Реалізовано два способи створення підпису: малювання мишею на Canvas та завантаження готового зображення-файлу. Усі звернення до СУБД виконуються через PDO з підготовленими виразами; паролі зберігаються у вигляді bcrypt-хешів. Вебдодаток розгорнуто в OpenServer 6.5.1 та протестовано функціональним методом «чорної скриньки» – 13 з 14 визначених тестових сценаріїв пройдено успішно, що відповідає покриттю функціональних вимог на рівні 92,8%. Техніко-економічне обґрунтування підтвердило економічну доцільність розробки: розрахунковий показник рентабельності інвестицій ROI склав близько 211%, термін окупності – менше одного року, що суттєво вигідніше порівняно з щорічними передплатами на комерційні аналоги.

Таким чином, поставлені завдання вирішено у повному обсязі. Напрямами подальших досліджень є: реалізація безпосереднього вбудовування зображення підпису в структуру PDF-файлу засобами PHP-бібліотек `setasign/fpdfi` та `setasign/fpdf`; впровадження кваліфікованого електронного підпису через інтеграцію з вітчизняними КНЕДП (зокрема АЦСК «Дія»); розширення функціоналу системи шляхом додавання email-сповіщень для підписантів, підтримки маршрутів підписання з послідовним порядком та хмарного розгортання; підвищення безпеки за рахунок впровадження двофакторної автентифікації та захисту від CSRF-атак.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кваліфікований електронний підпис (КЕП). Державна міграційна служба України. URL: <https://dmsu.gov.ua/faq/kvalifikovaniy-elektronnij-pidpis-ker.html> (дата звернення: 20.04.2026).
2. Як отримати електронний підпис? URL: <https://diia.gov.ua/faq/1> (дата звернення: 20.04.2026).
3. Інформаційна безпека та кібербезпека держави: навчальний посібник / Н. М. Титова, Н. М. Рідей, В. П. Настрадін, М. М. Присяжнюк, С. М. Мамченко, С. В. Артюх, Р. О. Яворська; за заг. ред. М. М. Присяжнюка. Київ: Видавництво Ліра-К, 2024. 224 с.
4. Е-документи, електронний підпис, довірчі послуги: основні зміни 2025 року. Юридична Газета. URL: <https://yur-gazeta.com/dumka-eksperta/edokumenti-elektronnij-pidpis-dovirchi-poslugi-osnovni-zmini-2025-roku.html> (дата звернення: 20.04.2026).
5. Коли та як використовувати факсиміле при підписанні документів. URL: <https://news.dtkt.ua/accounting/automation/63510-koli-ta-iak-vikoristovuvati-faksimile-pri-pidpisanni-dokumentiv> (дата звернення: 20.04.2026).
6. Електронний документообіг – тренд 2024 року. М.Е.Дос. URL: <https://medoc.ua/news/elektronnij-dokumentobg-trend-2024-roku> (дата звернення: 20.04.2026).
7. Впровадження електронного документообігу: проблеми та перспективи / І. Саламаха та ін. *Український інформаційний простір*. 2025. № 2(16). С. 206–218. URL: [https://doi.org/10.31866/2616-7948.2\(16\).2025.345631](https://doi.org/10.31866/2616-7948.2(16).2025.345631) (дата звернення: 20.04.2026).
8. Про електронні документи та електронний документообіг: Закон України від 22.05.2003 № 851-IV. URL: <https://zakon.rada.gov.ua/laws/show/851-15> (дата звернення: 20.04.2026).
9. Електронний документообіг для впевненого бізнесу. EDIN. URL: <https://edin.ua/zakon-ukrayini-pro-elektronni-dokumenti-ta-elektronnij-dokumentobig/> (дата звернення: 20.04.2026).

10. Про електронну ідентифікацію та електронні довірчі послуги: Закон України від 05.10.2017 № 2155-VIII. URL: <https://zakon.rada.gov.ua/laws/show/2155-19> (дата звернення: 20.04.2026).

11. ДСТУ 4145-2002. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевіряння. Київ: Держстандарт України, 2003. URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=68769 (дата звернення: 20.04.2026).

12. Оксютенко К. В. Правові засади розвитку електронного урядування в Україні в умовах європейської інтеграції. *Київський часопис права*. 2024. № 3. С. 163-170. DOI: <https://doi.org/10.32782/klj/2024.3.24> (дата звернення: 20.04.2026).

13. Який сервіс ЕДО підійде вашому бізнесу (огляд). Finance.ua. 2025. URL: <https://finance.ua/ua/goodtoknow/yakyi-servis-edo-pidiide-vashomu-biznesu-ohliad> (дата звернення: 20.04.2026).

14. ВЧАСНО.ЕДО: огляд, відгуки, рейтинг. MIISOFT. URL: <https://miisoft.com.ua/product/vchasno-edo/> (дата звернення: 20.04.2026).

15. DEALS: огляд, відгуки, аналоги. MIISOFT. URL: <https://miisoft.com.ua/product/deals/> (дата звернення: 20.04.2026).

16. Електронні підписи на базі DocuSign, PandaDoc, AdobeSign, InStaSign та їх чинність в Україні. Юридична Газета. 2021. URL: <https://yur-gazeta.com/publications/practice/informaciyne-pravo-telekomunikaciyi/elektronni-pidpisi-na-bazi-docusign-pandadoc-adobesign-instasign-ta-yih-chinnist-v-ukrayini.html> (дата звернення: 20.04.2026).

17. Вибір типу проєкту за методологією: Waterfall, Agile, Hybrid. e5.ua. 2024. URL: <https://e5.ua/uk/blogpost-2/vybir-typu-proyektu-za-metodologiyeyu-waterfall-agile-hybrid/> (дата звернення: 20.04.2026).

18. Що таке PHP? PHP це серверна мова програмування для веброзробки. URL: <https://site-konstruktor.com.ua/php> (дата звернення: 20.04.2026).

19. PDF.js – відкрита бібліотека JavaScript для перегляду PDF у браузері. URL: <https://products.fileformat.com/uk/pdf/javascript/pdfjs/> (дата звернення: 20.04.2026).

20. Open Server Panel – комплексний інструмент для локальної веброзробки. URL: <https://ospanel.io> (дата звернення: 20.04.2026).

21. ДСТУ 34.602-89. Інформаційна технологія. Технічне завдання на створення автоматизованої системи. URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=57803 (дата звернення: 20.04.2026).

22. Захист систем е-документообігу: головні критерії безпеки. InBase. 2024. URL: <https://inbase.com.ua/zahyst-system-e-dokumentooobigu-golovni-kryteriyi-bezpeky/> (дата звернення: 20.04.2026).

23. Технічне завдання на розробку сайту: як його скласти. URL: <https://webstudio2u.net/webdesign/737-tehnicheskoe-zadanie-na-razrabotku-saita.html?lang=ua> (дата звернення: 20.04.2026).

24. Про схеми баз даних. URL: <https://foxminded.ua/skhemy-bazy-danyh/> (дата звернення: 20.04.2026).

25. Що таке патерн MVC для програмування динамічних вебзастосунків. 2025. URL: <https://foxminded.ua/shho-take-patern-mvc-dlya-programuvannya-dynamichnyh-vebzastosunkiv/> (дата звернення: 20.04.2026).

26. Тестування інтерфейсу користувача: стандарти, чек-лист та кроки. URL: <https://wezom.com.ua/ua/blog/testing-ui-user-interface> (дата звернення: 20.04.2026).

27. Ключові UI принципи для створення якісного дизайну вашого веб продукту. URL: <https://turumburum.ua/blog/klyuchovi-ui-principi-dlya-stvorenniya-yakisnogo-dizaynu-vashogo-veb-produktu> (дата звернення: 20.04.2026).
11
головних трендів UX/UI у 2026 році: що потрібно знати дизайнерам і бізнесу. URL: <https://www.promodo.ua/blog/osnovni-trendi-ux-ui-dizaynu> (дата звернення: 20.04.2026).

29. Для чого потрібні UML діаграми?. URL: <https://foxminded.ua/uml-diagramy/> (дата звернення: 20.04.2026).

30. Як будувати UML-діаграми. Розбираємо три найпопулярніші варіанти. DOU. URL: <https://dou.ua/forums/topic/40575/> (дата звернення: 20.04.2026).