

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 1

ЗАТВЕРДЖЕНО

Науково-методичною радою
Державного університету
«Житомирська політехніка»
протокол від 03.04.2026р. № 2

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ для виконання завдань НАВЧАЛЬНОЇ ПРАКТИКИ

для здобувачів вищої освіти освітнього ступеня «БАКАЛАВР»
спеціальності F3 «Комп'ютерні науки»
освітньо-професійна програма «Комп'ютерна графіка та розробка ігор»
факультет інформаційно-комп'ютерних технологій
кафедра комп'ютерних наук

Рекомендовано на засіданні кафедри
комп'ютерних наук
«19» березня 2026 р.
протокол № 3

Розробники: старший викладач кафедри комп'ютерних наук МАРЧУК Дмитро
PhD, доцент кафедри комп'ютерних наук ЛЕВКІВСЬКИЙ Віталій
старший викладач кафедри комп'ютерних наук МАРЧУК Галина

Житомир
2026

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 2

Зміст

1. Загальні положення.....	3
1.1. Мета та завдання практики	3
1.3. Організація роботи	3
1.2. Стек технологій.....	4
2. Календарний план	5
2.1. Графік виконання (2 тижні)	5
3. Технічні вимоги до реалізації	6
3.1. Архітектура проєкту	6
3.2. Вимоги до Entity Framework (Code First)	6
3.3. Вимоги до WPF-інтерфейсу.....	7
3.4. Обов'язкові функції кожного проєкту	7
4. Перелік завдань (30 варіантів).....	9
5. Вимоги до звіту та захисту	13
5.1. Структура звіту	13
5.2. Захист проєкту	13
5.3. Критерії оцінювання.....	14
6. Рекомендації	15
6.1. Встановлення NuGet-пакетів	15
6.2. Типові помилки та їх виправлення	15
6.3. Корисні посилання.....	15
Додатки	22

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 3

1. Загальні положення

1.1. Мета та завдання практики

Навчальна практика є обов'язковою складовою освітньої програми для здобувачів вищої освіти освітнього ступеня «БАКАЛАВР» та спрямована на закріплення теоретичних знань з об'єктно-орієнтованого програмування (ООП) на практиці. Студенти набувають навичок розробки повноцінних настільних застосунків із застосуванням сучасного технологічного стеку.

Головна мета практики - у команді (самостійно) спроектувати, розробити та задокументувати програмний продукт, що включає графічний інтерфейс, бізнес-логіку та базу даних.

Завдання практики:

- опанувати розробку GUI-застосунків на платформі WPF (Windows Presentation Foundation);
- навчитися проектувати базу даних підходом Code First засобами Entity Framework;
- практично застосувати принципи ООП: інкапсуляцію, успадкування, поліморфізм;
- сформувати навички командної роботи та розподілу задач;
- скласти технічну документацію до розробленого програмного продукту.

1.2. Організація роботи

Практика виконується у командах по 3 студента. Кожна команда отримує одне завдання зі списку, узгоджує вибір із викладачем. Розподіл обов'язків між учасниками здійснюється самостійно учасниками групи.

Рекомендований розподіл ролей:

- *Team Lead / Архітектор* - проектування архітектури, моделі даних, фінальна інтеграція;

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 4

- *Backend розробник* - реалізація бізнес-логіки, Entity Framework, репозиторії;
- *Frontend розробник* - розробка XAML-інтерфейсу, прив'язка даних, UX.

Обов'язкові вимоги до організації:

- використання системи контролю версій Git із репозиторієм на GitLab;
- мінімум 5 комітів від кожного учасника протягом практики;
- щоденне оновлення репозиторію;
- ведення простого журналу задач.

1.3. Стек технологій

Для виконання практичних завдань використовується наступний стек технологій:

<i>Технологія</i>	<i>Версія</i>	<i>Призначення</i>
C# / .NET	.NET 8+	Мова програмування та платформа виконання
WPF	вбудований у .NET	Побудова графічного інтерфейсу (GUI)
Entity Framework Core	8.x (NuGet)	ORM для роботи з базою даних
SQLite	через EF Provider	Вбудована реляційна БД
Code First	підхід EF	Моделювання БД через C#-класи
Visual Studio	2022+	IDE для розробки
Git	будь-яка версія	Контроль версій та командна робота

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 5

2. Календарний план

2.1. Графік виконання (2 тижні)

День	Тиждень	Основний зміст роботи	Результат
1–2	1	Ознайомлення з завданням. Аналіз предметної області. Проектування бази даних (ER-діаграма). Визначення сутностей та зв'язків між ними.	ER-діаграма Додаток А
3	1	Налаштування проєкту Visual Studio. Встановлення NuGet-пакетів (EF Core, SQLite). Створення моделей (Entity-класів) та DbContext.	Структура проєкту Додаток Б
4	1	Перша міграція (Initial Migration). Налаштування DbContext. Seed-дані для тестування. Перевірка створення БД.	Міграція, БД
5	1	Реалізація шару доступу до даних: репозиторії або сервіси для кожної сутності. Тестування CRUD-операцій.	Репозиторії
6–7	1	Розробка бізнес-логіки. Реалізація основних функцій застосунку. Юніт-тестування логіки (за бажанням).	Бізнес-логіка
8–9	2	Розробка головного вікна (MainWindow). Навігація між вікнами / вкладками. Прив'язка даних (Binding) до колекцій.	Головне вікно
10–11	2	Розробка CRUD-форм: форми додавання, редагування, видалення записів. Валідація введення даних.	CRUD-форми
12	2	Реалізація пошуку, фільтрації та сортування. Додаткові функції згідно з описом завдання.	Пошук/фільтр
13	2	Фінальне налагодження (debugging). Обробка виняткових ситуацій (try/catch). Покращення UX.	Стабільний код
14	2	Підготовка документації, звіту, презентації. Фінальна демонстрація викладачу. Здача роботи.	Здача проєкту

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 6

3. Технічні вимоги до реалізації

3.1. Архітектура проєкту

Проєкт повинен мати чітку структуру папок, що відображає шарову архітектуру:

- Models/ - Entity-класи (сутності предметної області);
- Data/ - DbContext та конфігурації (OnModelCreating) (Додаток В);
- Repositories/ або Services/ - логіка роботи з даними;
- Views/ - XAML-файли вікон та UserControl;
- ViewModels/ - (за наявності MVVM) класи ViewModel;
- Helpers/ - допоміжні класи, утиліти, конвертери.

3.2. Вимоги до Entity Framework (Code First)

При реалізації шару даних необхідно дотримуватись наступних вимог:

1. Усі сутності мають мати первинний ключ (Id або НазваId типу int або Guid).
2. Навігаційні властивості обов'язкові для всіх зв'язків (один-до-багатьох, багато-до-багатьох).
3. Міграції повинні зберігатись у папці Migrations/ та бути в репозиторії.
4. DbContext ініціалізується один раз (Singleton або через using-блок).
5. Seed-дані (Initial data) - не менше 5–10 записів для демонстрації.
6. Рядок підключення до SQLite вказується у App.config або appSettings.

Приклад мінімальної моделі:

```
public class Product
{
    public int Id { get; set; }
    public string Name { get; set; } = string.Empty;
    public decimal Price { get; set; }
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/F2.00.Б/ОК35-2026
	Екземпляр № 1	Арк 26 / 7

```
public int CategoryId { get; set; }
public Category Category { get; set; } = null!;
}
```

3.3. Вимоги до WPF-інтерфейсу

Графічний інтерфейс повинен відповідати наступним вимогам:

- Головне вікно містить навігацію між розділами (TabControl або Menu або NavigationWindow).
- Основне відображення даних - DataGridView із можливістю вибору рядка.
- Форми введення/редагування - через окреме модальне вікно (ShowDialog).
- Обов'язкова валідація введення: перевірка на порожнє поле, числовий діапазон, формат дати.
- Повідомлення користувачу через MessageBox.Show() при помилках та успішних операціях.
- Кнопки мають бути логічно згруповані і мати зрозумілі підписи (Додати / Редагувати / Видалити / Пошук).
- Застосунок не повинен «зависати» при виконанні операцій з БД (async/await для довгих запитів).

3.4. Обов'язкові функції кожного проєкту

Незалежно від конкретного завдання, кожен проєкт повинен містити:

#	Функція	Критерії виконання
1	Перегляд (Read)	DataGridView відображає всі записи з БД
2	Додавання (Create)	Форма з валідацією, запис зберігається в SQLite
3	Редагування (Update)	Форма заповнена поточними даними, зміни зберігаються
4	Видалення (Delete)	Підтвердження через MessageBox, запис видаляється
5	Пошук/фільтр	Список оновлюється в реальному часі або за кнопкою
6	Обробка помилок	try/catch скрізь, де є операції з БД або введення
7	Seed-дані	При першому запуску БД заповнюється тестовими даними

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 8

Важливо пам'ятати!

Регулярно робіть коміти до Git - це страхує від втрати роботи і показує ваш реальний внесок.

Кожна функція - окремий коміт із зрозумілим описом: "Add Product CRUD forms" або "Fix validation in AddProductWindow".

Не чекайте до останнього дня - завантажуйте код щодня!

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/F2.00.Б/ОК35-2026
	Екземпляр № 1	Арк 26 / 9

4. Перелік завдань (30 варіантів)

Повний перелік варіантів завдань за рівнем складності:

№	Назва / Тема	Опис завдання	Рівень
1	Калькулятор (WPF + ООП)	Простий калькулятор з операціями +, -, *, /. Реалізувати через класи операцій із поліморфізмом (патерн Strategy). Підтримка десяткових чисел та обробка ділення на нуль.	Легкий
2	Записник (WPF + CRUD)	Застосунок для ведення нотаток з можливістю створення, редагування та видалення записів. Дані зберігати у SQLite через Entity Framework. Пошук за назвою.	Легкий
3	Конвертер одиниць (WPF + логіка)	Конвертер між одиницями вимірювання (температура, довжина, маса). Кожна категорія — окремий клас-конвертер. Інтерфейс з вибором категорії та полями введення.	Легкий
4	Список справ (To-Do) (WPF + EF + SQLite)	Менеджер задач із пріоритетами та статусами (нова/в роботі/виконана). Відображення у DataGrid із фільтрацією за статусом. Дані у SQLite.	Легкий
5	Словник (WPF + EF + SQLite)	Персональний словник іноземних слів із перекладом, прикладами і категоріями. Тренажер — випадкові слова для перевірки знань. Пошук і фільтр за категорією.	Легкий
6	Таймер і секундомір (WPF + ООП)	Додаток з таймером зворотнього відліку і секундоміром із проміжними часами (laps). Збереження сесій у SQLite з можливістю перегляду статистики.	Легкий
7	Телефонна книга (EF + SQLite + CRUD)	Контакти з іменем, телефоном, email, фото-URL та групами. Повний CRUD, пошук і сортування. Експорт списку у текстовий файл.	Легкий
8	Облік фінансів (EF + SQLite)	Менеджер особистих фінансів: доходи та витрати по категоріях, баланс, фільтр за датою. Зведена	Середній

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 10

		таблиця по місяцях. Різні типи транзакцій через успадкування.	
9	Каталог книг (EF + SQLite + зв'язки)	Бібліотечний каталог із книгами, авторами (зв'язок М:М) та жанрами. Пошук за автором/жанром/роком. Позначка «Прочитано» із датою.	Середній
10	Менеджер паролів (EF + SQLite + шифр.)	Зберігання облікових записів (сайт, логін, пароль). Генератор паролів із налаштуванням складності. Шифрування даних у БД. Майстер-пароль.	Середній
11	Щоденник харчування (EF + SQLite + звіти)	Трекер їжі: продукти з КБЖУ, прийоми їжі по датах, добові норми. Відображення прогресу відносно цілі. Статистика за тиждень у вигляді таблиці.	Середній
12	Облік автопарку (EF + SQLite + звіти)	База автомобілів: марка, модель, рік, пробіг, ТО. Журнал обслуговування. Повідомлення про наближення строку ТО. Пошук і фільтрація.	Середній
13	Магазин (POS) (EF + SQLite + бізнес)	Каса магазину: товари з цінами та залишками, кошик, оформлення продажу, зміна залишків. Чек із розрахунком здачі. Журнал продажів.	Середній
14	Аптека (EF + SQLite + бізнес)	Облік медикаментів: назва, виробник, кількість, термін придатності. Попередження про закінчення терміну. Пошук за назвою/виробником. Журнал надходжень.	Середній
15	Готель (EF + SQLite + зв'язки)	Система броні: кімнати різних категорій, гості, бронювання з датами. Перевірка доступності. Розрахунок вартості. Звіт по завантаженню.	Середній
16	Розклад занять (EF + SQLite + UI)	Розклад для студентів: предмети, викладачі, аудиторії, час. Перегляд по днях тижня. Перевірка конфліктів у розкладі. Фільтр за групою.	Середній
17	Клінічна картотека (EF + SQLite +	Картки пацієнтів: ПІБ, дата народження, діагнози, лікар, візити. Журнал прийомів. Пошук пацієнта. Перегляд усіх діагнозів пацієнта.	Середній

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 11

	<i>зв'язки)</i>		
18	HR-система (EF + SQLite + звіти)	Облік співробітників: відділи, посади, зарплати, дата прийому. Нарахування щомісячних зарплат. Звіт по відділах. Пошук і фільтрація.	Середній
19	Онлайн-бібліотека (EF + SQLite + MVVM)	Система видачі книг: книги, читачі, видача з терміном. Штраф за прострочення. Пошук по різних критеріях. Реалізувати MVVM патерн.	Складний
20	Планувальник проєктів (EF + SQLite + граф)	Проєкти з підзадачами, виконавцями та дедлайнами. Відображення задач у вигляді дерева (TreeView). Прогрес виконання. Сповіщення про дедлайн.	Складний
21	Система тестування (EF + SQLite + логіка)	Конструктор тестів: питання, варіанти відповідей, правильний варіант. Режим проходження тесту з таймером. Підрахунок балів, збереження результатів.	Складний
22	Склад (WMS) (EF + SQLite + звіти)	Складський облік: товари, постачальники, переміщення між зонами. Інвентаризація. Звіт по залишках. Попередження при мінімальній кількості.	Складний
23	Фітнес-трекер (EF + SQLite + графіки)	Тренування: вправи, підходи, ваги, дати. Програми тренувань. Графік прогресу (зростання ваг). Розрахунок загального об'єму навантаження.	Складний
24	Агентство нерухомості (EF + SQLite + MVVM)	Об'єкти нерухомості: тип, адреса, ціна, фото-URL, статус. Клієнти та угоди. Пошук за параметрами. Статистика по агентах.	Складний
25	Чат (локальний) (EF + SQLite + Timer)	Локальний чат для декількох користувачів (без мережі). Кімнати чату, повідомлення зі штампом часу. Оновлення через Timer. Сповіщення про нові повідомлення.	Складний
26	Система навчання	LMS: ролі (адмін/викладач/студент), курси, уроки, завдання, оцінки. Різний інтерфейс залежно від	Складний

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 12

	(EF + SQLite + ролі)	ролі. Прогрес проходження курсу.	
27	Аукціон (EF + SQLite + логіка)	Система торгів: лоти з описом і стартовою ціною, ставки учасників, таймер аукціону, визначення переможця. Журнал торгів. Статистика.	Складний
28	Турнірна таблиця (EF + SQLite + алгор.)	Менеджер спортивних турнірів: команди/учасники, матчі, рахунки, автоматичний розрахунок таблиці. Плей-оф bracket. Статистика гравців.	Складний
29	Банківська система (EF + SQLite + патерни)	Банківські рахунки різних типів (успадкування), перекази між рахунками, кредити з нарахуванням відсотків, виписка операцій. Патерни Repository + UoW.	Складний
30	ERP-система (EF + SQLite + модулі)	Комплексна система: склад, продажі, фінанси, HR як окремі модулі. Навігаційне меню. Спільна БД. Зведений дашборд із ключовими показниками.	Складний

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 13

5. Вимоги до звіту та захисту

5.1. Структура звіту

Звіт оформлюється у форматі .docx (.pdf) та має містити наступні розділи:

1. Титульна сторінка (назва роботи, номер варіанту, склад групи, дата).
2. Постановка задачі (короткий опис завдання та мети за стосунку).
3. Аналіз предметної області (опис сутностей і їх властивостей).
4. Проєктування бази даних (ER-діаграма із поясненням зв'язків).
5. Архітектура за стосунку (структура проєкту, шари, опис основних класів).
6. Опис реалізації (скріншоти застосунку з поясненням кожного вікна).
7. Код ключових класів (Entity-класи, DbContext, один репозиторій, одна форма).
8. Інструкція користувача (покроковий опис роботи з за стосунком).
9. Висновки (що було зроблено, які труднощі виникли, що засвоєно).
10. Посилання на репозиторій GitLab.

5.2. Захист проєкту

Захист відбувається в останній день практики у форматі демонстрації застосунку та відповідей на запитання. Кожен учасник команди повинен:

- продемонструвати особисто написану частину коду та пояснити її;
- відповісти на 2–3 теоретичних запитання за темою своєї частини;
- показати власні коміти в системі контролю версій.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 14

5.3. Критерії оцінювання

Критерій	Балів (макс.)	Деталі
Функціональність (CRUD + основні функції)	35	<i>Всі 7 обов'язкових функцій реалізовані</i>
Якість бази даних (Code First + EF)	20	<i>Коректні моделі, міграції, зв'язки</i>
Якість інтерфейсу WPF	15	<i>Зручність, валідація, UX</i>
Архітектура та якість коду	15	<i>Чіткий поділ шарів, ООП-принципи</i>
Командна робота (Git)	10	<i>Коміти від кожного, pull requests</i>
Звіт та захист	5	<i>Повнота документації, відповіді</i>

6. Рекомендації

6.1. Встановлення NuGet-пакетів

У Package Manager Console виконати наступні команди:

```
# Встановлення Entity Framework Core з SQLite
Install-Package Microsoft.EntityFrameworkCore.Sqlite
Install-Package Microsoft.EntityFrameworkCore.Tools

# Або через dotnet CLI
dotnet add package Microsoft.EntityFrameworkCore.Sqlite
dotnet add package Microsoft.EntityFrameworkCore.Tools

# Команди для міграцій
Add-Migration InitialCreate      # Створити першу міграцію
Update-Database                  # Застосувати міграцію
```

6.2. Типові помилки та їх виправлення

Типова помилка	Рішення
DbContext disposed занадто рано	Не зберігайте DbContext у полі класу між запитами. Використовуйте using() або DI.
Lazy loading не працює	Додайте virtual до навігаційних властивостей або використовуйте .Include().
Migration не бачить змін	Перевірте, що клас доданий у DbSet<T> у DbContext.
SQLite: таблиця не знайдена	Запустіть Update-Database. Перевірте, що рядок підключення вказує на правильний файл.
DataGrid не оновлюється	Використовуйте ObservableCollection<T> замість List<T> для Binding.
NullReferenceException у формі	Перевірте, що DataContext встановлено перед InitializeComponent().

6.3. Корисні посилання

- Документація Entity Framework Core:

<https://learn.microsoft.com/ef/core/>

- WPF Tutorial: <https://wpf-tutorial.com/>

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/F2.00.Б/ОК35-2026
	Екземпляр № 1	Арк 26 / 16

- SQLite Browser (перегляд БД): <https://sqlitebrowser.org/>
- Git Guide: <https://learngitbranching.js.org/>
- C# Best Practices: <https://learn.microsoft.com/dotnet/csharp/>

7. Чеклист перед задачею проєкту

Перед задачею роботи кожна команда повинна самостійно перевірити виконання всіх пунктів наведеного нижче чеклисту. Знак ✓ проставляється після перевірки. За кожний невиконаний пункт можуть знижуватись бали.

✓	Пункт перевірки	Блок	Балів
	Застосунок запускається без помилок	WPF	обов'язк.
	База даних (файл .db) створюється автоматично при першому запуску	EF Core	обов'язк.
	Є мінімум 1 міграція у папці Migrations/	EF Core	обов'язк.
	Є щонайменше 2 Entity-класи з навігаційними властивостями	Models	обов'язк.
	Усі 4 операції CRUD працюють (Create, Read, Update, Delete)	Функціонал	35
	Кнопка «Видалити» показує MessageBox із підтвердженням	WPF / UX	15
	Усі поля введення перевіряються перед збереженням (валідація)	WPF / UX	15
	Пошук або фільтрація за хоча б одним полем реалізовані	Функціонал	35
	Проєкт має шарову структуру: Models / Data / Services / Views	Архітектура	15
	Операції з БД обгорнуті у try/catch з повідомленням користувачу	Код	15
	DataGrid використовує ObservableCollection<T> для Binding	WPF	15
	Seed-дані присутні (щонайменше 5 тестових записів)	EF Core	обов'язк.
	Репозиторій GitLab містить мінімум 5 комітів від	Git	10

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 17

	кожного учасника		
	Файл .db відсутній у репозиторії (є у .gitignore)	Git	10
	Звіт містить усі 10 обов'язкових розділів	Звіт	5
	Кожен учасник може пояснити особисто написану частину коду	Захист	5

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/F2.00.Б/ОК35-2026
	Екземпляр № 1	Арк 26 / 18

8. Мінімальний робочий приклад

У цьому розділі наведено мінімальний набір файлів та коду, достатній для того, щоб застосунок запусився, відобразив порожню таблицю і дозволив додати перший запис. Цей приклад можна використовувати як «стартову точку» будь-якого варіанту завдання - достатньо замінити назви сутностей відповідно до свого варіанту.

8.1. Крок 1 - Entity-клас (Models/Item.cs)

```
namespace MyApp.Models;

public class Item
{
    public int Id { get; set; }
    public string Name { get; set; } = string.Empty;
    public string? Description { get; set; }
}
```

8.2. Крок 2 - DbContext (Data/AppDbContext.cs)

```
// Data/AppDbContext.cs
using Microsoft.EntityFrameworkCore;
using MyApp.Models;

public class AppDbContext : DbContext
{
    public DbSet<Item> Items { get; set; }
    protected override void OnConfiguring(DbContextOptionsBuilder opt)
        => opt.UseSqlite("Data Source=app.db");
}
```

8.3. Крок 3 - Міграція та запуск (Package Manager Console)

```
# 1. Встановити пакети
Install-Package Microsoft.EntityFrameworkCore.Sqlite
Install-Package Microsoft.EntityFrameworkCore.Tools

# 2. Створити першу міграцію
Add-Migration InitialCreate
# 3. Застосувати — файл app.db з'явиться автоматично
Update-Database
```

8.4. Крок 4 - Головне вікно (Views/MainWindow.xaml.cs)

```
public partial class MainWindow : Window
{
    private readonly AppDbContext _db;
    public MainWindow()
    {
        InitializeComponent();
        _db = new AppDbContext();
        _db.Database.Migrate(); // БД створюється автоматично
        Grid.ItemsSource = _db.Items.ToList();
    }
}
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 19

```
}  
private void Add_Click(object s, RoutedEventArgs e)  
{  
    _db.Items.Add(new Item { Name = NameBox.Text });  
    _db.SaveChanges();  
    Grid.ItemsSource = _db.Items.ToList();  
}  
}
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08-05.02/2/F2.00.Б/ОК35-2026
	Екземпляр № 1	Арк 26 / 20

9. Глосарій

Рекомендується ознайомитися з визначеннями перед початком практики.

Термін	Визначення
ORM	Object-Relational Mapping - технологія, що дозволяє працювати з базою даних через об'єкти C# без написання SQL-запитів вручну.
Entity Framework Core	ORM-бібліотека від Microsoft для .NET, яка автоматично перетворює C#-класи на таблиці бази даних і навпаки.
Code First	Підхід у EF Core, при якому розробник спочатку описує структуру даних у вигляді C#-класів, а EF автоматично генерує схему бази даних через міграції.
DbContext	Клас у EF Core, що представляє сеанс з'єднання з базою даних. Містить DbSet<T> для кожної сутності і відповідає за виконання запитів та збереження змін.
Migration (міграція)	Автоматично згенерований файл із кодом, що описує зміни схеми бази даних (створення таблиць, додавання колонок тощо). Застосовується командою Update-Database.
CRUD	Абревіатура чотирьох базових операцій з даними: Create (створення), Read (читання), Update (оновлення), Delete (видалення).
WPF	Windows Presentation Foundation - платформа .NET для створення настільних застосунків з графічним інтерфейсом. Інтерфейс описується мовою розмітки XAML.
XAML	Extensible Application Markup Language - мова розмітки на основі XML, якою описується інтерфейс WPF-застосунку (кнопки, таблиці, вікна тощо).
Binding (прив'язка даних)	Механізм WPF, що автоматично синхронізує значення властивостей UI-елементів із даними (наприклад, DataGrid відображає колекцію об'єктів без ручного заповнення рядків).
ObservableCollection	Колекція, яка автоматично сповіщає DataGrid або інший

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 21

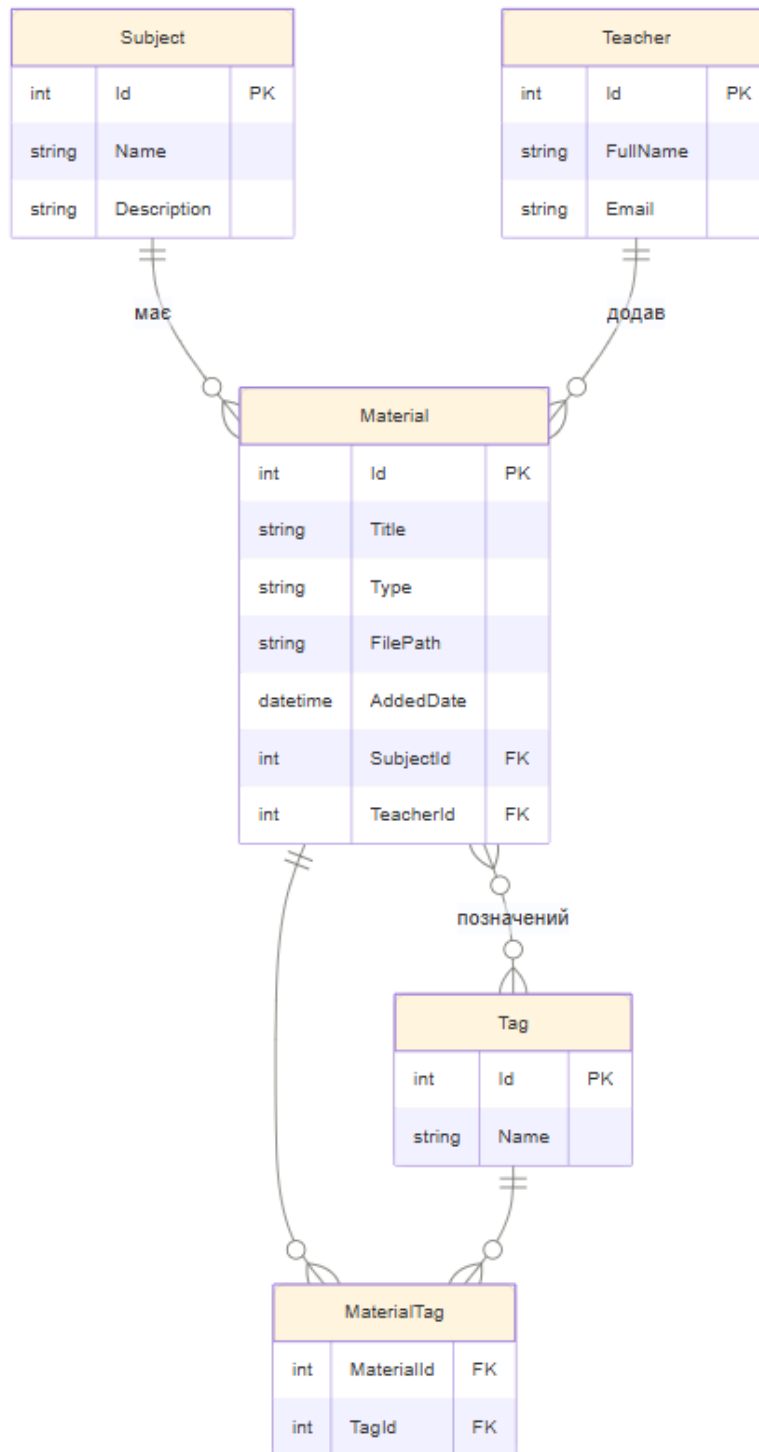
	елемент про зміни (додавання, видалення елементів), що призводить до миттєвого оновлення відображення.
SQLite	Вбудована реляційна база даних, що зберігає всі дані в одному файлі (.db). Не потребує встановлення окремого сервера - ідеальна для навчальних проєктів.
LINQ	Language Integrated Query - синтаксис запитів у C#, що дозволяє фільтрувати, сортувати та перетворювати колекції даних (у тому числі таблиці БД через EF Core).
Seed-дані	Початкові (тестові) дані, що автоматично додаються до бази даних при першому запуску міграцій через метод HasData() у OnModelCreating.
Навігаційна властивість	Властивість Entity-класу, що посилається на пов'язану сутність або колекцію сутностей. EF Core використовує її для автоматичного побудови JOIN-запитів.
Merge Request	Запит на злиття гілки у GitLab. Дозволяє іншим учасникам переглянути зміни та залишити коментар перед тим, як код потрапить до основної гілки.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 22

Додатки

Додаток А

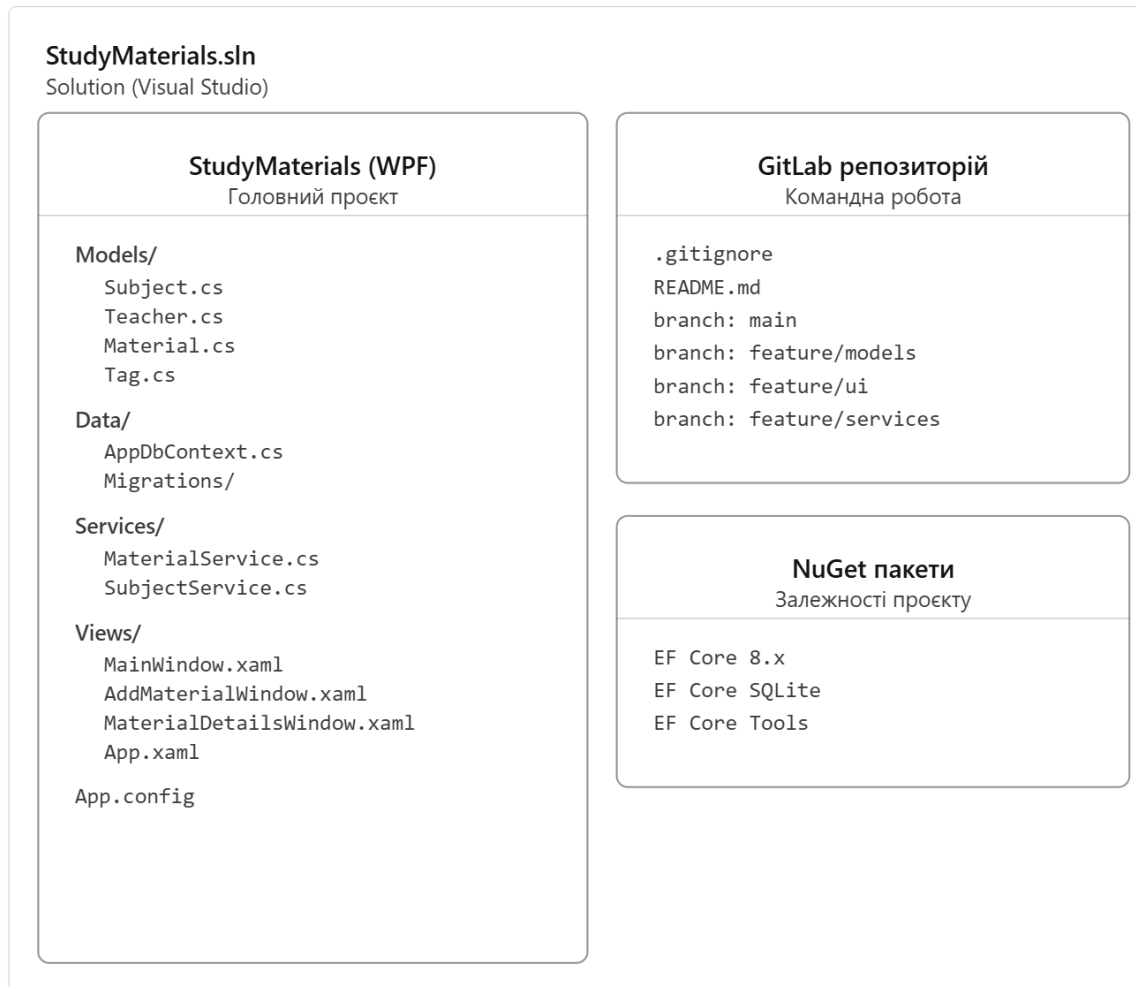
Етап 1. Аналіз предметної області та ER-діаграма, поет: «Облік навчальних матеріалів»



Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 24

Додаток Б

Етап 2. Структура проєкту та GitLab



Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 25

Додаток В

Етап 4. DbContext та Seed-дані

```
// Data/AppDbContext.cs

using Microsoft.EntityFrameworkCore;
using StudyMaterials.Models;

namespace StudyMaterials.Data;

public class AppDbContext : DbContext
{
    public DbSet<Subject> Subjects { get; set; }
    public DbSet<Teacher> Teachers { get; set; }
    public DbSet<Material> Materials { get; set; }
    public DbSet<Tag> Tags { get; set; }

    protected override void OnConfiguring(DbContextOptionsBuilder options)
        => options.UseSqlite("Data Source=study_materials.db");

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        // M:M — EF Core 5+ через тінюву таблицю
        modelBuilder.Entity<Material>()
            .HasMany(m => m.Tags)
            .WithMany(t => t.Materials)
            .UsingEntity(j => j.ToTable("MaterialTags"));

        // Seed-дані — предмети
        modelBuilder.Entity<Subject>().HasData(
            new Subject { Id = 1, Name = "Програмування C#", Description = "ООП та .NET" },
            new Subject { Id = 2, Name = "Бази даних", Description = "SQL, EF Core" },
            new Subject { Id = 3, Name = "Web-розробка", Description = "HTML, CSS, JS" }
        );

        // Seed-дані — викладачі
        modelBuilder.Entity<Teacher>().HasData(
            new Teacher { Id = 1, FullName = "Іваненко О.М.", Email = "ivanenko@ua" },
            new Teacher { Id = 2, FullName = "Петренко В.С.", Email = "petrenko@ua" }
        );

        // Seed-дані — теги
        modelBuilder.Entity<Tag>().HasData(
            new Tag { Id = 1, Name = "ООП" },
            new Tag { Id = 2, Name = "SQLite" },
            new Tag { Id = 3, Name = "WPF" },
            new Tag { Id = 4, Name = "LINQ" }
        );
    }
}
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.08- 05.02/2/F2.00.Б/ОК35- 2026
	Екземпляр № 1	Арк 26 / 26

Додаток Г

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра комп'ютерних наук

Звіт
з навчальна практика

на тему: «**Автоматичне виділення ключових понять і термінів з тексту**»

студентів 1 курсу групи КН-25-1
спеціальність ФЗ «Комп'ютерні науки»

Груницького Дмитра Сергійовича

(прізвище, ім'я та по-батькові)

Груницького Дмитра Сергійовича

(прізвище, ім'я та по-батькові)

Груницького Дмитра Сергійовича

(прізвище, ім'я та по-батькові)

Керівник **ст. викл. каф. КН Марчук Г.В.**

Дата захисту: «__» _____ 20__ р.

Національна оцінка: _____

Кількість балів: _____

Оцінка: ECTS _____

Члени комісії:

(підпис)

(підпис)

(підпис)

Віталій ЛЕВКІВСЬКИЙ

(прізвище та ініціали)

Галина МАРЧУК

(прізвище та ініціали)

Дмитро МАРЧУК

(прізвище та ініціали)

Житомир – 2026