

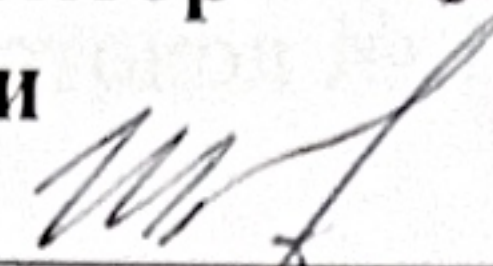
ЗАКЛАД ВИЩОЇ ОСВІТИ
«УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА»

Факультет суспільних і прикладних наук

Кафедра інформаційних технологій

ЗАТВЕРДЖУЮ

Проректор з навчально-методичної роботи

 Ярослав ШТАНЬКО
“31” 008 2026 р.

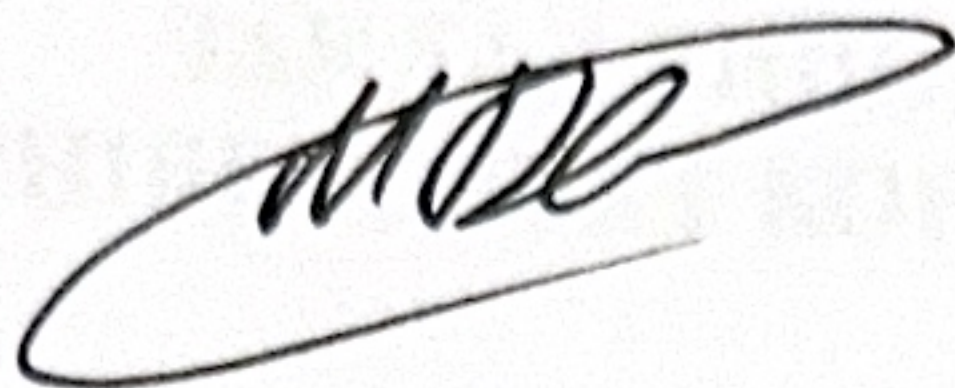
Prompt-інжиніринг

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Галузь знань:	12 Інформаційні технології
Спеціальність:	121 Інженерія програмного забезпечення
Освітньо-професійна (освітньо-наукова) програма:	Розробка та тестування програмного забезпечення
Освітній рівень:	перший (бакалаврський)
Статус дисципліни:	обов'язкова
Мова викладання, навчання та оцінювання:	українська
Семестр викладання	V

Івано-Франківськ
2026

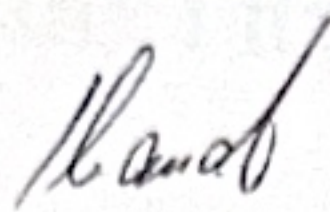
РОЗРОБНИК:
доцент кафедри ІТ



Микола ДЕМЧИНА

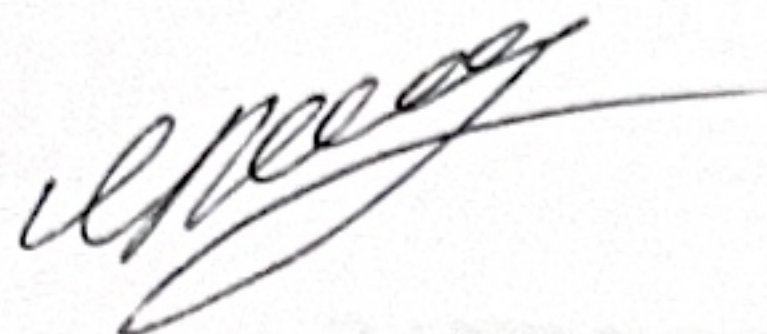
ЗАТВЕРДЖЕНО:
на засіданні кафедри інформаційних технологій, протокол № 1 від 27.08.2026 р.

Завідувач кафедри
УЗГОДЖЕНО:



Олександр ІВАНОВ

Гарант ОПП/ОНП



Володимир МАКОВИШИН

СХВАЛЕНО:
на засіданні Науково-методичної ради, протокол № 1 від 31.08.2026 р.

e-mail	mykola.demchyna@ukd.edu.ua
Номер аудиторії чи кафедри	206
Посилання на сайт	https://ukd.edu.ua
Сторінка курсу в СДО	https://online.ukd.edu.ua/

ВСТУП

Анотація навчальної дисципліни

Навчальна дисципліна «Prompt-інжиніринг» формує системний підхід до проєктування, перевірки та супроводу інструкцій для великих мовних і мультимодальних моделей. Курс охоплює будову промпту, роботу з контекстом і прикладами, структуровані відповіді, використання зовнішніх інструментів, оцінювання якості, захист від ін'єкцій і відповідальне застосування генеративного штучного інтелекту в інженерії програмного забезпечення.

Навчання поєднує лекції, демонстрації та практичні експерименти. Здобувачі створюють версії промптів для вимог, коду, тестів і документації, формують набори перевірок, порівнюють результати за визначеними критеріями та документують обмеження. Усі результати моделі розглядаються як гіпотези, що потребують перевірки людиною та за допомогою програмних засобів.

Мета навчальної дисципліни – сформувати здатність розробляти відтворювані, керовані та безпечні промпти й робочі процеси на їх основі для завдань інженерії програмного забезпечення, оцінювати їхню якість на даних і відповідально інтегрувати мовні моделі в програмні системи.

Для досягнення мети поставлені такі основні **завдання**:

- зрозуміти принципи роботи й обмеження великих мовних моделей та вплив контексту на результат;
- опанувати техніки постановки завдання, використання прикладів, декомпозиції та визначення формату відповіді;
- навчитися будувати набори перевірок, критерії та метрики для систематичного порівняння промптів;
- застосовувати захист від prompt injection, правила приватності та принципи відповідального використання ШІ.

У результаті вивчення навчальної дисципліни студент повинен **знати**:

- складники промпту, ієрархію інструкцій, межі контексту та основні параметри генерації;
- техніки zero-shot і few-shot, декомпозицію задач, структуровані відповіді, grounding та базові принципи RAG;
- підходи до побудови еталонних наборів, рубрик, автоматизованих перевірок і версіонування промптів;
- ризики галюцинацій, упереджень, витоку даних, prompt injection і надмірної довіри до результатів моделі.

У результаті вивчення навчальної дисципліни студент повинен **уміти**:

- формулювати мету, контекст, обмеження, приклади та критерії приймання для промпту;
- отримувати й програмно перевіряти структуровані відповіді, проєктувати виклики зовнішніх інструментів;
- створювати контрольний набір, проводити порівняльне тестування та аналізувати типові помилки;
- виявляти ін'єкції та небезпечні дані, розмежовувати довірені інструкції і недовірений вміст;

Компетентності та результати навчання, яких набувають здобувачі освіти внаслідок вивчення навчальної дисципліни (шифри та зміст компетентностей та програмних результатів навчання вказано відповідно до ОПП “Розробка та тестування програмного забезпечення”.

Шифр та назва компетентності	Шифр та назва програмних результатів навчання
ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.	ПРН2. Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності.
ЗК2. Здатність застосовувати знання у практичних ситуаціях.	ПРН9. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.
ЗК8. Здатність діяти на основі етичних міркувань.	ПРН13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.
СК1. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.	ПРН19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.
СК3. Здатність розробляти архітектури, модулі та компоненти програмних систем.	
СК14. Здатність до алгоритмічного та логічного мислення.	

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Курс	3		
Семестр	5		
Кількість кредитів ECTS	3		
Аудиторні навчальні заняття		денна форма	заочна форма
	лекції	14	2
	практичні	28	6
Самостійна робота		48	82
Форма підсумкового контролю	Залік		

Структурно-логічна схема вивчення навчальної дисципліни

Пререквізити	Постреквізити
Основи роботи з нейронними мережами; Основи програмування; Інженерія програмного забезпечення.	Документування програмних продуктів; Якість програмного забезпечення і тестування.

Для практичних занять потрібні браузер, редактор коду, доступ до СДО та щонайменше один дозволений викладачем сервіс або API генеративного ШІ. Секретні ключі зберігаються поза кодом; персональні, конфіденційні та службові дані не передаються до зовнішніх систем.

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Перелік тем лекційного матеріалу

Тема 1. Великі мовні моделі та завдання prompt-інжинірингу (2 год.)

Принцип передбачення наступного токена, контекстне вікно, імовірнісний характер відповіді, параметри генерації. Можливості й межі моделей у завданнях аналізу вимог, програмування, тестування та документування.

Питання для самостійного вивчення: Порівняти поняття prompt design, prompt engineering і context engineering; описати типові джерела нестабільності результатів. Література: [1, 2, 3].

Тема 2. Структура промпту та ієрархія інструкцій (2 год.)

Мета, вхідні дані, контекст, роль, обмеження, критерії приймання та формат відповіді. Системні, розробницькі й користувацькі інструкції. Розділювачі, шаблони та змінні.

Питання для самостійного вивчення: Побудувати шаблон промпту з явними контрактами входу й виходу для задачі інженерії ПЗ. Література: [3, 5, 6].

Тема 3. Приклади, декомпозиція та керування міркуванням (2 год.)

Zero-shot і few-shot prompting. Добір позитивних і негативних прикладів. Поділ складної задачі, уточнювальні запитання, самоперевірка та незалежна верифікація без покладання на приховані міркування моделі.

Питання для самостійного вивчення: Підібрати приклади для класифікації вимог і створити план перевірки відповіді. Література: [1, 3, 5].

Тема 4. Структуровані відповіді та зовнішні інструменти (2 год.)

JSON Schema, машинно-читані контракти, валідація й повторні спроби. Виклик функцій, інструментів і API; межа між текстом моделі та детермінованою логікою програми.

Питання для самостійного вивчення: Спроектувати схему відповіді та правила валідації для перетворення вимоги на тест-кейси. Література: [3, 7, 8].

Тема 5. Контекст, grounding, RAG і мультимодальні промпти (2 год.)

Відбір, очищення та структурування контексту. Посилання на джерела, робота з довгими документами, основи retrieval-augmented generation. Текстові, графічні та файлові входи.

Питання для самостійного вивчення: Скласти промпт для відповіді за наданими джерелами з правилом відмови за браку доказів. Література: [3, 5, 6].

Тема 6. Оцінювання, тестування та версіонування промптів (2 год.)

Цільова поведінка, еталонний набір, рубрики, точні перевірки й модельні оцінювачі. Порівняння версій, аналіз помилок, регресійне тестування, вартість, затримка та відтворюваність.

Питання для самостійного вивчення: Підготувати мінімальний набір тестів і рубрику для одного промптового сценарію. Література: [4, 6, 9].

Тема 7. Безпека, приватність, етика та життєвий цикл (2 год.)

Прямі й непрямі prompt injection, витік інструкцій і даних, надмірні повноваження, упередження та галюцинації. Людський контроль, журналювання, моніторинг, управління змінами та критерії відмови від автоматизації.

Питання для самостійного вивчення: Провести загрозове моделювання промптового застосування й запропонувати багаторівневі захисні заходи. Література: [7, 9, 10].

Зміст практичних занять

Заняття 1. Базове дослідження поведінки моделі (2 год.)

Фіксація моделі, параметрів і версії промпту; повторні запуски; аналіз варіативності, галюцинацій і невизначеності.

Заняття 2. Перепроєктування нечіткої інструкції (2 год.)

Перетворення неоднозначного запиту на промпт із метою, контекстом, обмеженнями, критеріями приймання та форматом відповіді.

Заняття 3. Zero-shot і few-shot класифікація вимог (2 год.)

Класифікація функціональних і нефункціональних вимог; добір прикладів; перевірка на граничних і неоднозначних випадках.

Заняття 4. Декомпозиція задачі та перевірка проміжних результатів (2 год.)

Розбиття складної задачі на етапи, визначення контрактів між етапами, запровадження уточнень і незалежних перевірок.

Заняття 5. Структурований вивід і JSON Schema (2 год.)

Формування тест-кейсів за заданою схемою; валідація типів, обов'язкових полів і допустимих значень; оброблення помилок.

Заняття 6. Виклик інструментів у програмному сценарії (2 год.)

Проектування опису функції, аргументів і правил виклику; перевірка дозволів; відокремлення рішення моделі від виконання дії.

Заняття 7. Grounding на документах (2 год.)

Створення відповіді лише за наданими фрагментами; цитування джерел; визначення ситуацій, коли доказів недостатньо.

Заняття 8. Мультимодальний промпт (2 год.)

Аналіз макета, діаграми або скриншота разом із текстовими вимогами; зіставлення спостережень із критеріями якості.

Заняття 9. Набір перевірок і рубрика оцінювання (2 год.)

Побудова репрезентативного набору вхідних даних, еталонів і рубрики; розмежування точних, евристичних і експертних перевірок.

Заняття 10. Порівняльне тестування версій промпту (2 год.)

А/В-порівняння, аналіз помилок, вартості й затримок; фіксація рішення щодо прийняття або відхилення версії.

Заняття 11. Захист від prompt injection (2 год.)

Моделювання прямих і непрямих ін'єкцій; розмежування інструкцій і даних; валідація виходу та мінімізація повноважень.

Заняття 12. Промпти для розроблення та рев'ю коду (2 год.)

Формування вимог до коду, тестів і рев'ю; перевірка через компіляцію, статичний аналіз і тести; аналіз хибних рекомендацій.

Заняття 13. Промптовий робочий процес для інженерії ПЗ (2 год.)

Поєднання шаблонів, структурованого виходу, інструментів, журналювання та контрольного набору в один відтворюваний процес.

Заняття 14. Захист підсумкового проєкту (2 год.)

Демонстрація сценарію, тестового набору, результатів і ризиків; взаємне рецензування; план подальшого моніторингу.

Зміст самостійної роботи здобувачів**Розподіл годин, виділених на вивчення дисципліни:**

Найменування видів робіт	Розподіл годин	
	денна форма	заочна форма
Самостійна робота, год, у т.ч.:	48	82
Опрацювання матеріалу, викладеного на лекціях	7	20
Підготовка до практичних занять та контрольних заходів	10	8
Підготовка та оформлення звітів, журналу експериментів і портфоліо	8	-
Підготовка до поточного та підсумкового контролю	5	-
Опрацювання матеріалу, винесеного на самостійне вивчення, та виконання підсумкового проєкту	18	54

ПОЛІТИКА КУРСУ

Процедура проведення контрольних заходів, а саме поточного контролю знань протягом семестру та підсумкового семестрового контролю, регулюється «Положенням про систему поточного та підсумкового контролю оцінювання знань та визначення рейтингу здобувачів освіти». Ознайомитись з документом можна за [ПОКЛИКАННЯМ](#).



Здобувачі мають право оскаржувати результати поточного оцінювання, а також інформувати про факти конфліктних ситуацій в академічній групі чи з викладачем у відповідності до «Положення про політику та врегулювання конфліктних ситуацій». Ознайомитись з документом можна за [ПОКЛИКАННЯМ](#).



Згідно “Положення про організацію освітнього процесу” здобувач допускається до семестрового контролю з конкретної навчальної дисципліни (семестрового екзамену, диференційованого заліку), якщо він виконав усі види робіт, передбачені на семестр навчальним планом та силабусом/робочою програмою навчальної дисципліни, підтвердив опанування на мінімальному рівні результатів навчання (отримав ≥ 35 бали), відпрацював визначені індивідуальним навчальним планом всі лекційні, практичні, семінарські та лабораторні заняття, на яких він був відсутній. Ознайомитись з документом можна за [ПОКЛИКАННЯМ](#).



Вимоги щодо дотримання академічної доброчесності, види її порушень і порядок притягнення до академічної відповідальності визначено Положенням про систему забезпечення академічної доброчесності в Університеті Короля Данила. Ознайомитись з документом можна за [ПОКЛИКАННЯМ](#).



Використання систем штучного інтелекту під час виконання навчальних завдань допускається відповідно до “Політики використання штучного інтелекту в Університеті Короля Данила”. Здобувач зобов’язаний критично перевіряти створений ШІ зміст, нести відповідальність за подану роботу та зазначати факт і спосіб використання таких систем. Ознайомитись з документом можна за [ПОКЛИКАННЯМ](#).



Використання мобільних телефонів, планшетів та інших гаджетів під час лекційних та практичних занять дозволяється виключно у навчальних цілях (для уточнення певних даних, перевірки правопису, отримання довідкової інформації тощо). На гаджетах повинен бути активований режим «без звуку» до



початку заняття. Під час занять заборонено надсилання текстових повідомлень, прослуховування музики, перевірка електронної пошти, соціальних мереж тощо. Під час виконання заходів контролю використання гаджетів заборонено (за винятком, коли це передбачено умовами його проведення). У разі порушення цієї заборони результат анулюється без права перескладання. Ознайомитись з відповідним документом можна за [ПОСИЛАННЯМ](#).

Комунікація відбувається через електронну пошту і сторінку дисципліни в Moodle.

Процедури визнання результатів навчання, здобутих шляхом формальної/інформальної освіти визначаються «Положенням про порядок визнання результатів навчання, здобутих шляхом неформальної та / або інформальної освіти». Ознайомитись з документом можна за [ПОКЛИКАННЯМ](#).



МЕТОДИ НАВЧАННЯ

При вивченні дисципліни застосовується комплекс методів для організації навчання студентів з метою розвитку їх логічного та абстрактного мислення, творчих здібностей, підвищення мотивації до навчання та формування особистості майбутнього фахівця в галузі інформаційних технологій.

Результат навчання	Метод навчання	Метод оцінювання
ПРН2. Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності.	Аналіз кейсів професійної відповідальності, мінілекція, обговорення ризиків і меж автоматизації.	Ситуаційні завдання, аналіз ризиків у портфоліо, комп'ютерне тестування.
ПРН9. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.	Демонстрація структури промπτу, практичне формулювання й уточнення вимог, взаємне рецензування.	Перевірка промπτів і критеріїв приймання, захист практичних робіт, тестування.
ПРН13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.	Практикуми з промπτами, декомпозиція задач, структуровані відповіді, інтеграція інструментів і програмна валідація.	Перевірка відтворюваності, аналіз коду й результатів, підсумковий проєкт, тестування.
ПРН19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.	Побудова контрольного набору, рубрик і регресійних перевірок; аналіз помилок і порівняння версій.	Звіт про оцінювання, захист проєкту, завдання з верифікації та валідації, тестування.

ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Вид	Зміст	% від загальної оцінки	Бал	
			min	max
Поточні контрольні заходи	всього	60	35	60
Підсумкові контрольні заходи	диференційований залік	40	25	40
Всього:	-	100	60	100

Процедура проведення контрольних заходів, а саме поточного контролю знань протягом семестру та підсумкового семестрового контролю, регулюється «Положенням про систему поточного та підсумкового контролю оцінювання знань та визначення рейтингу студентів».

Фіксація поточного контролю здійснюється в “Електронному журналі обліку успішності академічної групи” на підставі чотирибальної шкали - “2”; “3”; “4”; “5”. У разі відсутності студента на занятті виставляється “н”. За результатами поточного контролю у Журналі, автоматично визначається підсумкова оцінка, здійснюється підрахунок пропущених занять.

Усі пропущені заняття, а також негативні оцінки студенти зобов'язані відпрацювати впродовж трьох наступних тижнів. У випадку недотримання цієї норми, замість “н” в журналі буде виставлено “0” (нуль балів), без права перездачі. Відпрацьоване лекційне заняття в електронному журналі позначається літерою «в».

До підсумкового контролю допускаються студенти які за результатами поточного контролю отримали не менше 35 балів. Усі студенти, що отримали 34 балів і менше, не допускаються до складання підсумкового контролю і на підставі укладання додаткового договору, здійснюють повторне вивчення дисципліни впродовж наступного навчального семестру.

За результатами підсумкового контролю (диференційований залік/екзамен) студент може отримати 40 балів.

Студенти, які під час підсумкового контролю отримали 24 бали і менше, вважаються такими, що не здали екзамен/диференційований залік і повинні йти на перездачу.

Загальна семестрова оцінка з дисципліни, яка виставляється в екзаменаційних відомостях оцінюється в балах (згідно Шкали оцінювання знань за ЄКТС) і є сумою балів отриманих під час поточного та підсумкового контролю.

Підсумковий контроль з дисципліни Prompt-інжиніринг проводиться у вигляді Залік.

Критерії оцінювання практичних робіт і портфоліо

Оцінка	Критерії
«2» – незадовільно	Промпт не відповідає завданню або результат не перевірено; відсутні істотні дані для відтворення; здобувач не може пояснити рішення чи його ризики.
«3» – задовільно	Основний сценарій працює за типовим зразком; структура промπτу й перевірка часткові; контрольний набір вузький; пояснення потребує уточнень.
«4» – добре	Рішення відповідає вимогам у стандартних і кількох граничних сценаріях; наведено критерії, тести й аналіз помилок; результати відтворювані.
«5» – відмінно	Рішення самостійне й обґрунтоване; контрольний набір репрезентативний; порівняно версії, вартість і ризики; вихід валідовано, а обмеження чітко задокументовано.

Шкала оцінювання знань за ЄКТС

Оцінка за національною шкалою	Рівень досягнень, бали	Шкала ECTS
Національна диференційована шкала		
Відмінно	90 – 100	A
Добре	83 – 89	B
	75 – 82	C
Задовільно	67 – 74	D
	60 – 66	E
Незадовільно	35 – 59	FX
	0 – 34	F
Національна недиференційована шкала		
Зараховано	60 – 100	-
Не зараховано	0 – 59	-

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

Основна література

1. White J., Fu Q., Hays S. et al. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. 2023. URL: <https://arxiv.org/abs/2302.11382> (дата звернення: 13.09.2026).
2. Liu P., Yuan W., Fu J. et al. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. ACM Computing Surveys. 2023. Vol. 55, No. 9. URL: <https://arxiv.org/abs/2107.13586> (дата звернення: 13.09.2026).
3. White J., Hays S., Fu Q. et al. ChatGPT Prompt Patterns for Improving Code Quality, Refactoring, Requirements Elicitation, and Software Design. 2023. URL: <https://arxiv.org/abs/2303.07839> (дата звернення: 13.09.2026).
4. NIST. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile. NIST AI 600-1. 2024. URL: <https://doi.org/10.6028/NIST.AI.600-1> (дата звернення: 13.09.2026).

Електронні інформаційні ресурси

5. OpenAI. Prompting. Official API documentation. URL: <https://developers.openai.com/api/docs/guides/prompting> (дата звернення: 13.09.2026).
6. OpenAI. Working with evals. Official API documentation. URL: <https://developers.openai.com/api/docs/guides/evals> (дата звернення: 13.09.2026).
7. OpenAI. Structured model outputs. Official API documentation. URL: <https://developers.openai.com/api/docs/guides/structured-outputs> (дата звернення: 13.09.2026).
8. Anthropic. Prompt engineering overview. Claude Platform Docs. URL: <https://platform.claude.com/docs/en/build-with-claude/prompt-engineering/overview> (дата звернення: 13.09.2026).
9. Google Cloud. Overview of prompting strategies. Generative AI on Vertex AI. URL: <https://cloud.google.com/vertex-ai/generative-ai/docs/learn/prompts/prompt-design-strategies> (дата звернення: 13.09.2026).
10. OWASP GenAI Security Project. LLM01:2025 Prompt Injection. URL: <https://genai.owasp.org/llmrisk/llm01-prompt-injection/> (дата звернення: 13.09.2026).