

УДК 378.12:004.8

**Тетяна Острецова**

провідний концертмейстер кафедри музичного мистецтва та хореографії  
Луганський національний університет імені Тараса Шевченка, Полтава, Україна  
*t.axenova@gmail.com*  
ORCID: [0009-0001-9584-0365](https://orcid.org/0009-0001-9584-0365)

**Олена Пономаренко**

концертмейстер кафедри музичного мистецтва та хореографії  
Луганський національний університет імені Тараса Шевченка, Полтава, Україна  
*elenapon91@gmail.com*  
ORCID: [0009-0004-9053-2701](https://orcid.org/0009-0004-9053-2701)

**Анна Єзерська**

концертмейстер кафедри музичного мистецтва та хореографії  
Луганський національний університет імені Тараса Шевченка, Полтава, Україна  
*ezerskaorcid@gmail.com*  
ORCID: [0009-0000-4203-7571](https://orcid.org/0009-0000-4203-7571)

## АВТОМАТИЗАЦІЯ ЗВІТНОСТІ ВИКЛАДАЧА ЗАСОБАМИ ШТУЧНОГО ІНТЕЛЕКТУ ТА NO-CODE ПЛАТФОРМИ

**Анотація.** У сучасних закладах вищої освіти проблема надмірного адміністративного навантаження набуває критичного значення. Часто виникає феномен «цифрового парадоксу», коли впровадження централізованих інформаційних систем замість полегшення роботи лише збільшує час, витрачений на звітність. У дослідженні запропоновано розв'язання цієї проблеми через створення та емпіричну перевірку персональної автоматизованої системи обліку професійної діяльності викладача. Методологічну основу роботи становлять підхід Design Science Research та концепція «користувач-розробник» (Citizen Developer), яка дозволяє освітянам без спеціальної технічної підготовки створювати власні цифрові інструменти. Запропонована архітектура інтегрує месенджер Telegram, платформу оркестрації сценаріїв Make.com та мультимодальну модель штучного інтелекту Google Gemini 2.5 Flash. Розроблений алгоритм забезпечує розгалужену маршрутизацію різнорідних вхідних даних: текстових описів, голосових повідомлень, вебпосилань, зображень сертифікатів чи афіш. Система виконує семантичний аналіз контенту, вилучає ключові дати, генерує описи українською мовою та автоматично вносить записи до Google Sheets із гіперпосиланням на цифрові копії документів. Експериментальна апробація на базі факультету мистецтв Луганського національного університету імені Тараса Шевченка (м. Полтава) засвідчила скорочення часу фіксації одного досягнення в середньому з 4 хвилин 31 секунди до 14 секунд (економія часу – 95%). У статті проаналізовано технічні аспекти налаштування системних запитів (промптів) для досягнення точності класифікації понад 90%, а також розглянуто питання інформаційної безпеки. Результати доводять, що використання ШІ та доступних No-Code рішень дозволяє перетворити рутинні бюрократичні процедури на ефективні процеси, вивільняючи ресурси викладачів для науково-творчої діяльності.

**Ключові слова:** автоматизація звітності; користувач-розробник; citizen developer; Low-Code/No-Code; штучний інтелект; Google Gemini; Make.com

### ВСТУП

**Постановка й обґрунтування актуальності проблеми.** Сучасна вища освіта функціонує в умовах перманентної трансформації, спричиненої глобальними технологічними зрушеннями та локальними викликами. На тлі очевидних змін у змісті та формах навчання менш помітною, але критичною для академічної спільноти, постає проблема зростання «адміністративного тягаря». Як зазначають P. Woelert та ін., це явище не обмежується окремими бюрократичними процедурами, а є системним навантаженням, що знижує ефективність університету (Woelert et al., 2025). Емпіричні

дослідження фіксують тривожну тенденцію: частка робочого часу, що витрачається на адміністративний супровід, не зменшується. J. Schiller та S. LeMire наголошують, що вона набуває складніших форм, сягаючи 44-50% у періоди інтенсивної звітності (Schiller & LeMire, 2023). Адміністративні функції дедалі частіше перетворюються на основну, хоча й приховану складову навантаження викладача.

Виникає явище, яке в науковій літературі окреслюють як «цифровий парадокс». Впровадження централізованих інформаційних систем, покликаних оптимізувати управління, часто спричиняє зворотний ефект для кінцевого користувача. Замість очікуваного полегшення викладач отримує додаткові обов'язки оператора баз даних. Архітектура університетських систем зазвичай орієнтована на потреби адміністративного контролю та агрегації статистики, ігноруючи ергономіку роботи окремого доцента чи професора. Особливо гостро ця проблема постає в українських ЗВО, де рейтингування є стандартом кадрової політики. Процес збирання підтверджень активності, від публікації статті до проведення профорієнтаційної бесіди, залишається фрагментованим. Викладач змушений вручну акумулювати різноманітні докази (посилання, PDF-файли, фото дипломів, знімки екрана) та вносити їх у громіздкі таблиці. Це спричиняє когнітивний дисонанс: висококваліфікований фахівець витрачає години на механічну роботу, яку здатен виконати алгоритм початкового рівня.

Водночас, стрімкий розвиток генеративного штучного інтелекту та платформ Low-Code/No-Code (LCNC) відкриває принципово нові можливості. Нові підходи дають змогу користувачам без IT-підготовки самостійно розробляти цифрові рішення під власні потреби. Актуальність цього дослідження полягає в емпіричній перевірці ефективності таких персональних систем у реальному академічному середовищі. Метою роботи є не лише демонстрація технічної можливості автоматизації, але й обґрунтування нової ролі викладача в цифрову епоху – ролі користувача-розробника, творця власних екосистем, здатних мінімізувати бюрократичний тиск.

**Аналіз наукових досліджень і публікацій.** У сучасній науці концепція адміністративного тягаря розглядається через призму трьох типів витрат, яких зазнає громадянин (у нашому випадку – викладач) при взаємодії з системою (Moynihan et al., 2014):

1. Витрати на навчання: час та зусилля, необхідні для того, щоб розібратися, як працює система звітності, які показники змінилися в новому навчальному році та як правильно оформити документи. В умовах постійних змін нормативної бази ці витрати є значними.

2. Витрати відповідності: безпосередній час на збір довідок, сканування, перейменування файлів та внесення даних. Це найбільш очевидна частина «паперової роботи».

3. Психологічні витрати: стрес, фрустрація та відчуття втрати професійної автономії через необхідність постійно доводити свою ефективність формальними методами, що призводить до вигорання (Davidovitch & Cohen, 2024).

Наше дослідження пропонує інструментальний підхід до мінімізації саме витрат відповідності та психологічних витрат шляхом делегування рутини автоматизованій системі.

Донедавна автоматизація була ефективною лише для чітко структурованих даних. Однак, як вказують V. Bhardwaj та M. Kumar, інтеграція ШІ у сценарії LCNC-платформ докорінно змінює ситуацію. Сучасні моделі (такі як Google Gemini або GPT-5) виступають у ролі «інтелектуального прошарку»: вони здатні опрацьовувати неструктурований текст, «бачать» зображення та приймають рішення в умовах невизначеності (Bhardwaj & Kumar, 2025). У такій конфігурації сценарій автоматизації

стає простором взаємодії користувача та ШІ, а LCNC-платформа – організаційною основою цього процесу.

Питання зручності використання таких платформ детально розглянуто в огляді J. Pinho, P. Aguiar та L. Amaral. Автори підкреслюють привабливість LCNC-середовищ для нефахівців завдяки їх наочності, проте зауважують обмеженість їхніх можливостей при роботі зі змістом інформації (Pinho et al., 2022). Сама платформа є лише технічним каркасом. Для змістової інтерпретації даних (текстів, зображень) необхідне залучення зовнішнього інтелекту. Саме такий підхід ми використовуємо: генеративний ШІ залучається не як вбудована опція платформи, а як ключовий елемент сценарію, що виконує аналітичні завдання.

Практичні аспекти впровадження цих інструментів освітянами досліджували S. McHugh, N. Carroll та C. Connolly. Вони дійшли висновку, що вчителі готові до інновацій, але часто стримуються складністю технологій або браком часу. Автори наголошують: недостатньо просто надати доступ до програмного забезпечення. Необхідна зміна парадигми – від пасивного споживання до активного творення власних рішень (McHugh et al., 2023).

Створення програмних продуктів фахівцями, які не є професійними програмістами, у світовій практиці отримало назву Citizen Development. Згідно з визначенням Gartner, Citizen Developer – це співробітник, який створює бізнес-додатки для потреб організації, використовуючи санкціоновані інструменти. Системний аналіз взаємодії Citizen Development, LCNC-платформ та генеративного ШІ подано у праці J. Sodano та J. DeFranco. Автори розглядають демократизацію розробки як відповідь на дефіцит ІТ-кадрів і доводять, що сучасні інструменти знижують поріг входу в технології (Sodano & DeFranco, 2025). Особливу увагу вони приділяють галузевим експертам, які здатні самостійно автоматизувати свої робочі процеси. Водночас наголошується на ризиках (якість рішень, безпека) та доцільності гібридних моделей із чіткими правилами контролю. Ці висновки підтверджують перспективність обраного нами шляху автоматизації звітності.

У вітчизняному науковому дискурсі термінологія для опису цього явища ще формується. Англomовний термін Citizen Developer часто перекладають як «громадянський розробник» або «бізнес-розробник». Останній варіант невиправдано звужує сферу до комерції. У контексті освіти, науки та мистецтва, де метою є генерація знань і творчість, прив'язка до «бізнесу» виглядає штучною. Тому ми пропонуємо використовувати термін «користувач-розробник». Це визначення точно відбиває подвійну природу суб'єкта: він залишається активним користувачем, але набуває спроможності створювати власні інструменти, де технологія є лише засобом реалізації професійних потреб.

Водночас, проблематика зміни ролі викладача в умовах цифровізації розглядається й українськими дослідниками. Колектив авторів у складі М. Умрик, Н. Морзе та Є. Смирнової-Трибульської аргументує необхідність розширення цифрових компетентностей освітян до специфічних «AI-компетентностей» (Умрик та ін., 2025). Це передбачає не лише пасивне використання готових інструментів, а й здатність розуміти принципи роботи алгоритмів та адаптувати їх під власні потреби. Такий підхід створює теоретичне підґрунтя для концепції викладача як активного розробника власних цифрових рішень.

Важливим кроком у дослідженні інструментарію такої розробки стала робота В. Гужви, присвячена цифровій трансформації університетів через впровадження інтелектуальних агентів. Автор обґрунтував доцільність використання Low-Code платформ для створення «розумних» асистентів, здатних автоматизувати адміністративні процеси. Дослідник зазначає, що інтеграція штучного інтелекту в

корпоративні системи дозволяє делегувати програмним агентам рутинні завдання з обробки даних, що раніше виконувалися вручну (Гужва, 2024). Економічне обґрунтування цього підходу розширює у своїй праці О. Івасенко, наводячи дані про те, що LCNC-рішення дозволяють скоротити витрати на розробку на 40-70% та прискорити впровадження у 5-10 разів (Івасенко, 2025). Це доводить, що автоматизація силами викладачів є економічно вигідною стратегією.

Аналіз існуючих досліджень виявив певну нестачу знань у сфері практичного застосування LCNC-рішень для потреб освітян. Більшість наукових праць фокусуються або на дидактичних аспектах використання ШІ у навчанні студентів, або на глобальних питаннях цифровізації освіти. При цьому поза увагою дослідників залишається проблема створення локальних, персоналізованих інструментів, які б дозволили окремому викладачу автоматизувати власну звітну рутину, мінімізуючи бюрократичне навантаження.

**Мета дослідження** – теоретично обґрунтувати та експериментально перевірити модель персональної автоматизованої системи обліку професійної активності викладача («цифрового асистента»), створеної на базі доступних No-Code інструментів та генеративного ШІ, як засобу мінімізації адміністративного навантаження.

Для досягнення мети виконано такі **завдання**:

1. Розробити архітектуру системи.
2. Створити робочий прототип без написання програмного коду, використовуючи платформу No-Code.
3. Розробити систему запитів (промптів) для моделі ШІ, що забезпечить коректну класифікацію педагогічних та мистецьких досягнень.
4. Провести порівняльний аналіз часових витрат на звітування в ручному та автоматизованому режимах.
5. Оцінити економічну доступність запропонованого рішення.

## РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

В основу розробки покладено методологічний підхід Design Science Research, який передбачає створення та оцінку практичного артефакту (програмного рішення) для вирішення конкретної проблеми реального світу. У нашому випадку проблемою є надмірні часові витрати на звітність, а артефактом – автоматизований чат-бот.

Вибір технологічного стека був зумовлений кількома критеріями: доступність, тобто наявність безкоштовних тарифних планів, простота інтеграції та підтримка мобільних сценаріїв використання.

Як інтерфейс взаємодії було обрано месенджер Telegram (Рис. 1). Для українського освітнього простору це де-факто стандарт оперативної комунікації. Він не потребує встановлення додаткового ПЗ, працює стабільно навіть за умови слабого інтернет-з'єднання та дозволяє легко передавати файли різних форматів. Використання Telegram Bot API є безкоштовним, що усуває фінансові бар'єри.

Як ядро оркестрації процесів обрано платформу Make.com. Платформа при цьому не виконує аналітичних функцій, а забезпечує логіку виклику та взаємодії зовнішніх сервісів, зокрема генеративного ШІ. Її візуальний редактор дозволяє будувати логіку обробки даних у вигляді блок-схем, що є інтуїтивно зрозумілим. До того ж модульна структура Make дозволяє легко замінювати компоненти, наприклад, змінити Google Drive на Dropbox без переписування всього сценарію.

Інтелектуальним центром системи стала модель ШІ Google Gemini 2.5 Flash. Вибір саме цієї моделі продиктований її мультимодальними можливостями (здатністю одночасно аналізувати текст, аудіо та зображення) та високою швидкістю роботи при

низькій вартості. Крім того, наявність безкоштовного ліміту в Google AI Studio робить це рішення економічно привабливим для індивідуального використання.



Рис. 1. Загальна архітектура персональної автоматизованої системи обліку професійної активності викладача  
(Джерело: ілюстрацію створено авторами самостійно)

Емпіричною базою дослідження став аналіз реального масиву даних звітності викладачів факультету мистецтв ДЗ «ЛНУ імені Т. Шевченка» (афіші, програми концертів, новини, сертифікати конференцій, наукові статті, посібники).

Розроблена система становить собою складний, розгалужений алгоритм, реалізований у середовищі Make.com (Рис. 2). На відміну від простих лінійних чат-ботів, наш асистент здатний розпізнавати тип контенту, який надсилає користувач, і динамічно обирати відповідний сценарій обробки. Розглянемо логіку роботи основних компонентів системи.



Рис. 2. Сценарій в середовищі Make.com для обробки різних типів вхідних даних  
(Джерело: AI-асистент викладача, 2025)

Процес починається з модуля Telegram: Watch Updates, який працює через механізм Webhook. Для налаштування модуля необхідно звернутися до екосистеми Telegram. Процедура реєстрації бота є повністю безкоштовною та здійснюється через системного бота @BotFather. Результатом цієї дії є отримання унікального токена (API Token), який слугує ключем авторизації та дозволить модулю Telegram: Watch Updates отримувати повідомлення від користувача. Далі вступає в дію критично важливий елемент – маршрутизатор (Router). Його функція подібна до роботи залізничної стрілки: він

аналізує склад повідомлення і спрямовує його в одну з п'яти незалежних гілок. Це дозволяє уникнути конфліктів та оптимізувати кожен гілку під певну задачу.

**Службові команди.** У Telegram спілкування часто починається зі службових команд (наприклад, /start). Ми свідомо відокремили ці технічні сигнали від інтелектуальної обробки. Якщо повідомлення починається зі слешу «/», система спрямовує його в обхід штучного інтелекту. Це зроблено з двох причин: по-перше, щоб не витратити ресурси нейромережі на прості привітання, а по-друге – щоб запобігти виникненню «галюцинацій», коли ШІ намагається проаналізувати команду. Натомість користувач миттєво отримує чітку інструкцію-підказку, що роз'яснює алгоритм подання звіту. Це робить систему дружньою навіть для тих колег, які вперше з нею стикаються.

**Аналіз тексту.** Якщо повідомлення містить лише текст і не є посиланням чи командою, активується перша гілка. Текст передається безпосередньо в модель Gemini. Тут відбувається семантичний аналіз: ШІ порівнює довільний опис від викладача (наприклад, «вчора була опонентом у Іваненко») з офіційним класифікатором рейтингових показників університету (Положення про рейтингування, 2024), що інтегрований у системну інструкцію (System Prompt). Модель визначає відповідний код (наприклад, 2.7 – Офіційне опонування на захисті дисертації), вилучає дату та формулює лаконічний опис українською мовою (Рис. 3).

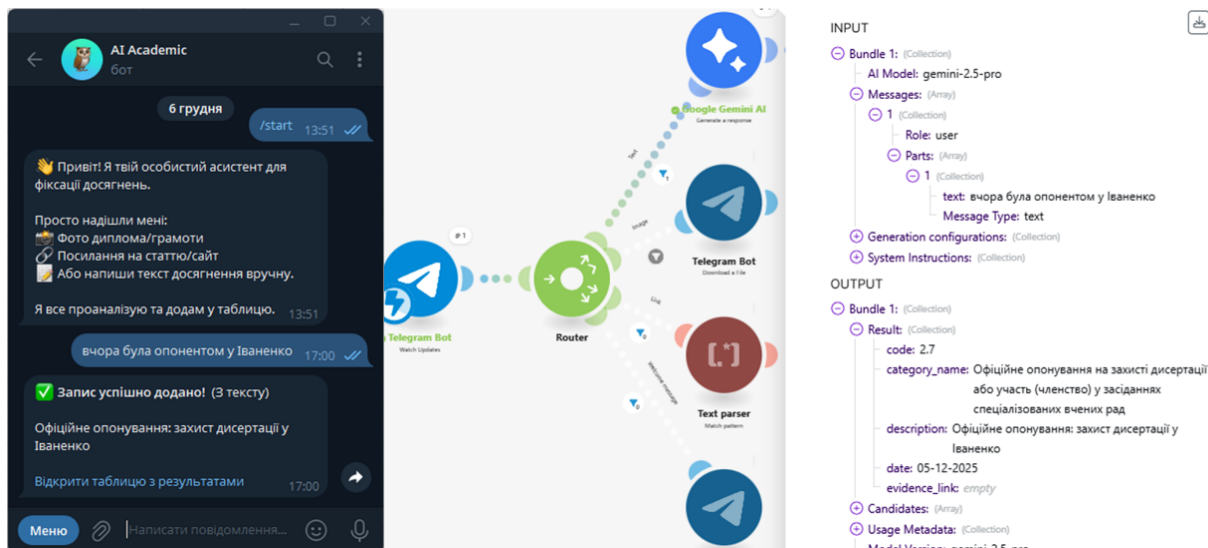


Рис. 3. Приклад обробки текстового повідомлення та класифікація досягнення за допомогою моделі Google Gemini 2.5 Flash (Джерело: ілюстрацію створено авторами самостійно)

Для використання моделі Gemini 2.5 Flash необхідно створити проєкт у середовищі Google AI Studio та згенерувати ключ API. Відповідь моделі у форматі JSON розподіляється по коміркам Google-таблиці. Таблицю необхідно попередньо створити і надати доступ до неї відповідному модулю системи Make.com.

**Аналіз зображень.** Іноді доказом досягнення є зображення, а саме – фото афіші або диплома. Коли маршрутизатор виявляє зображення, запускається наступна послідовність дій:

1. Бот завантажує файл максимальної якості з серверів Telegram.
2. Файл автоматично зберігається в папку на диску Google Drive, щоб створити постійне гіперпосилання-доказ.

3. Файл передається в модель Gemini, яка не просто зчитує текст, а аналізує візуальний контекст. Модель здатна відрізнити диплом лауреата від сертифіката

учасника, або афіші. Отримана інформація, як і в попередньому випадку, структурується та вноситься в Google-таблицю (Рис. 4).

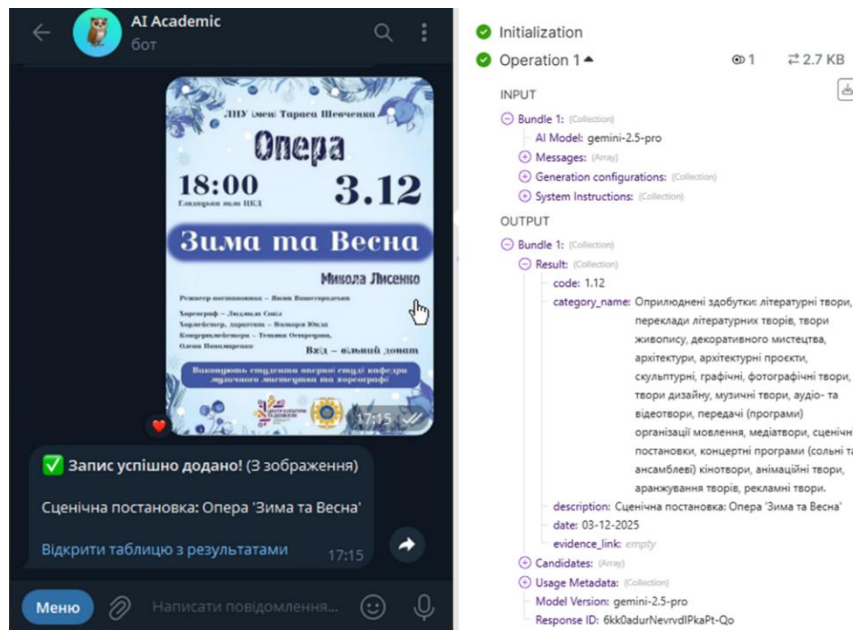


Рис. 4. Приклад обробки зображення у системі автоматизованого обліку (Джерело: ілюстрацію створено авторами самостійно)

Окрему увагу слід приділити налаштуванню доступу до файлового сховища Google Drive для персональних облікових записів (@gmail.com). У зв'язку з посиленими заходами безпеки Google, просте підключення сторонніх сервісів може бути недоступно. Для легального та безоплатного подолання цих обмежень треба створити власний проєкт у консолі Google Cloud Platform. У налаштуваннях проєкту необхідно активувати Google Drive API та налаштувати екран згоди OAuth (OAuth Consent Screen) у режим «External» зі статусом «Testing». Це дозволяє додати власну електронну адресу до списку тестових користувачів і отримати персональні ідентифікатори Client ID та Client Secret.

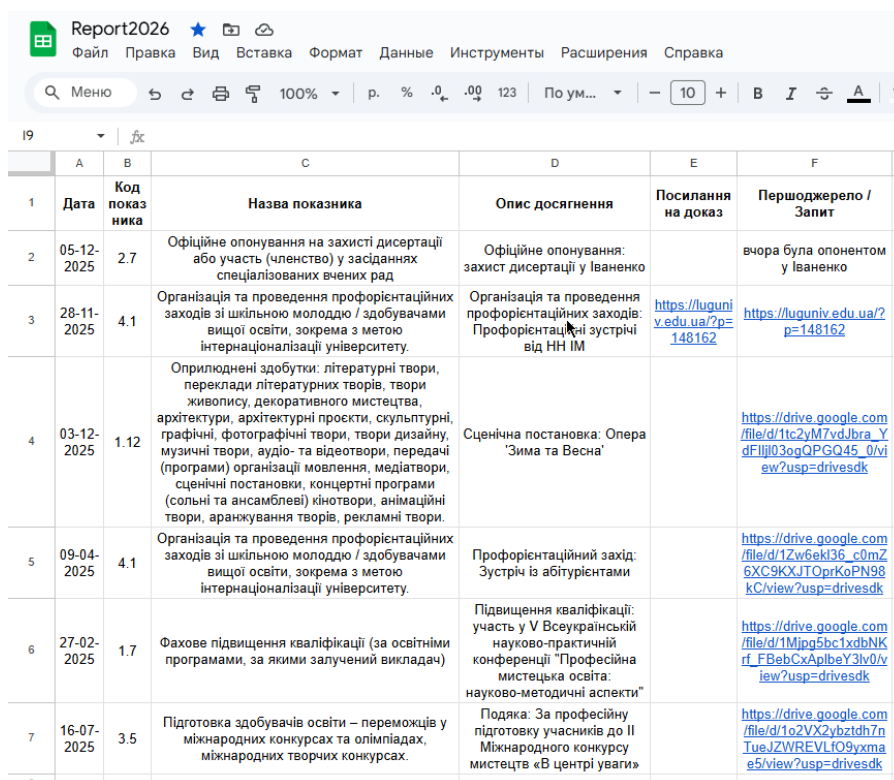
Слід зазначити, що під час налаштування доступу до Google Drive API у режимі тестування для зовнішніх користувачів система накладає обмеження на тривалість дії токенів оновлення – вони залишаються чинними лише протягом семи днів. У практичному вимірі це означає, що приблизно раз на тиждень робота сервісу автоматично переривається, а користувач змушений повторно виконувати процедуру входу до системи. З огляду на те, що така регулярна ручна авторизація суперечить ідеї автоматизації, доцільним є переведення додатку у Google Cloud Console в режим «In Production» без проходження формальної процедури верифікації від Google. Хоча під час входу користувач і отримує стандартне попередження про безпеку, цей крок дозволяє використовувати довготривалий токен доступу для особистих потреб. У результаті система може функціонувати безперервно, без необхідності щотижневого втручання з боку викладача, що істотно підвищує її практичну цінність та зручність у повсякденній роботі.

Така процедура, хоч і виглядає дещо технічно складнішою за інші, виконується одноразово, швидко, є безоплатною і надає повний контроль над безпекою власних даних, уникаючи необхідності проходити верифікацію додатка, яка вимагається для публічних комерційних продуктів.

Проте, якщо налаштування власного проєкту в Google Cloud Console для роботи з диском Google Drive викликає труднощі або помилки, користувач-розробник може легко

замінити цей блок наприклад на модуль Dropbox або Microsoft OneDrive. Зазначені сервіси мають значно простішу модель авторизації в середовищі Make, яка зазвичай реалізується в один клік через стандартне вікно входу без необхідності генерації складних ідентифікаторів клієнта. У такому випадку логіка сценарію залишається незмінною: система так само отримуватиме файл і генеруватиме посилання для звіту, змінюється лише місце фізичного зберігання даних.

Подібна варіативність характерна й для бази даних: замість таблиці Google Sheets (рис. 5) можуть використовуватися Airtable або Notion. Аналогічну гнучкість демонструє й інтелектуальне ядро системи: замість Google Gemini рішення легко адаптується до моделей OpenAI (GPT-5) або Anthropic (Claude). Це підтверджує, що в межах підходу «користувач-розробник» вирішальним є не вибір конкретного інструменту, а розуміння та проектування процесів, у яких викладач виступає конструктором власного цифрового рішення.



	A	B	C	D	E	F
1	Дата	Код показника	Назва показника	Опис досягнення	Посилання на доказ	Першоджерело / Запит
2	05-12-2025	2.7	Офіційне опонування на захисті дисертації або участь (членство) у засіданнях спеціалізованих вчених рад	Офіційне опонування: захист дисертації у Іваненко		вчора була опонентом у Іваненко
3	28-11-2025	4.1	Організація та проведення профорієнтаційних заходів зі шкільною молоддю / здобувачами вищої освіти, зокрема з метою інтернаціоналізації університету.	Організація та проведення профорієнтаційних заходів: Профорієнтаційні зустрічі від НН ІМ	<a href="https://luguniv.edu.ua/?p=148162">https://luguniv.edu.ua/?p=148162</a>	<a href="https://luguniv.edu.ua/?p=148162">https://luguniv.edu.ua/?p=148162</a>
4	03-12-2025	1.12	Оприлюднені здобутки: літературні твори, переклади літературних творів, твори живопису, декоративного мистецтва, архітектурні проекти, скульптурні, графічні, фотографічні твори, твори дизайну, музичні твори, аудіо- та відеотвори, передачі (програми) організації мовлення, медіатвори, сценічні постановки, концертні програми (сольні та ансамблеві) кіновтори, анімаційні твори, аранжування творів, рекламні твори.	Сценічна постановка: Опера 'Зима та Весна'		<a href="https://drive.google.com/file/d/1tc2yM7vdJbra_YdFijj03ogQPQQ45_0/vi ew?usp=drivesdk">https://drive.google.com/file/d/1tc2yM7vdJbra_YdFijj03ogQPQQ45_0/vi ew?usp=drivesdk</a>
5	09-04-2025	4.1	Організація та проведення профорієнтаційних заходів зі шкільною молоддю / здобувачами вищої освіти, зокрема з метою інтернаціоналізації університету.	Профорієнтаційний захід: Зустріч із абітурієнтами		<a href="https://drive.google.com/file/d/1Zw6ek136_c0mZ6XC9KXJT0prKoPN98kC/view?usp=drivesdk">https://drive.google.com/file/d/1Zw6ek136_c0mZ6XC9KXJT0prKoPN98kC/view?usp=drivesdk</a>
6	27-02-2025	1.7	Фахове підвищення кваліфікації (за освітніми програмами, за якими залучений викладач)	Підвищення кваліфікації: участь у V Всеукраїнській науково-практичній конференції "Професійна мистецька освіта: науково-методичні аспекти"		<a href="https://drive.google.com/file/d/1Mjpg5bc1xibnNKrf_FBebCxApIbeY3lv0/vi ew?usp=drivesdk">https://drive.google.com/file/d/1Mjpg5bc1xibnNKrf_FBebCxApIbeY3lv0/vi ew?usp=drivesdk</a>
7	16-07-2025	3.5	Підготовка здобувачів освіти – переможців у міжнародних конкурсах та олімпіадах, міжнародних творчих конкурсах.	Подяка: За професійну підготовку учасників до II Міжнародного конкурсу мистецтв «В центрі уваги»		<a href="https://drive.google.com/file/d/1o2VX2ybtzdh7nTueJZWREVLfO9yxmae5/view?usp=drivesdk">https://drive.google.com/file/d/1o2VX2ybtzdh7nTueJZWREVLfO9yxmae5/view?usp=drivesdk</a>
8						

Рис. 5. Зафіксовані у Google Sheets досягнення з посиланнями на цифрові копії документів  
(Джерело: ілюстрацію створено авторами самостійно)

**Аналіз вебресурсу.** Проста передача в модель ШІ посилання на статтю чи новину неефективна, оскільки моделі можуть не мати доступу до сторінки в реальному часі. Тому у нашій системі реалізовано механізм «скрейпінгу» (Рис. 6) (автоматизованого збору даних).

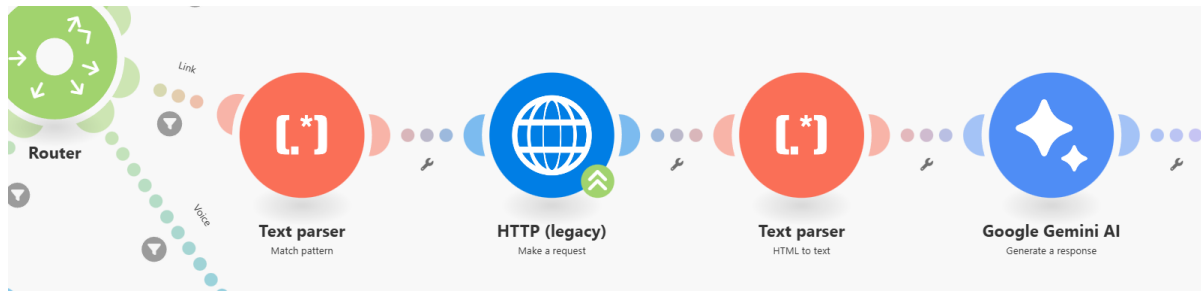


Рис. 6. Механізм «скрепінгу»  
(Джерело: AI-асистент викладача, 2025)

Система вилучає URL, робить HTTP-запит, отримує HTML-код сторінки та очищує його від технічного коду (тегів, скриптів). Лише чистий текст вебсторінки передається в Gemini для аналізу. Це підвищує ймовірність того, що до звіту потраплять лише корисні дані, що мінімізує ризик виникнення галюцинацій моделі.

**Аналіз голосових повідомлень.** Ще один спосіб передати інформацію про досягнення – це описати його власними словами, усно. В цій гілці система детектує наявність голосового повідомлення. Аудіофайл завантажується та передається в Gemini. Модель отримує інструкцію не просто транскрибувати мову в текст, а одразу аналізувати зміст. Це дозволяє викладачу диктувати звіти у вільному стилі, наприклад: «12 грудня, зустрічалася з випускниками полтавської школи №20. Спілкувалися на тему вибору майбутньої професії» (Рис. 7).

Система автоматично розпізнає дату, подію, класифікує це як 4.1 – «Організація та проведення профорієнтаційних заходів зі шкільною молоддю» і вносить запис до таблиці. Такий підхід радикально знижує когнітивний бар'єр входу: звітність стає такою ж простою, як розмова з колегою.

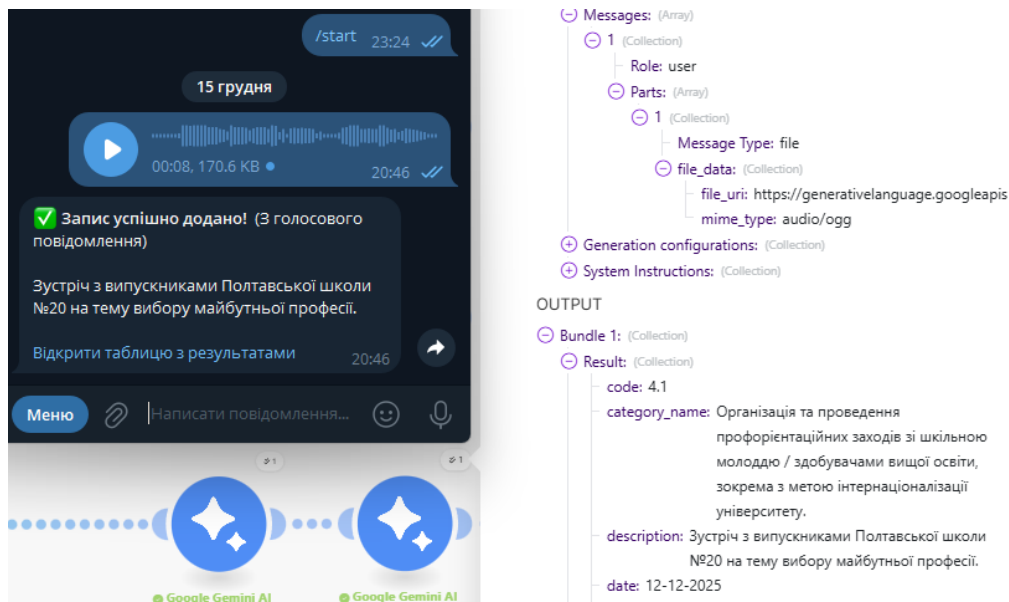


Рис. 7. Приклад обробки голосового повідомлення  
(Джерело: ілюстрацію створено авторами самостійно)

**Системні інструкції.** Незважаючи на відсутність потреби у програмуванні, автоматизація сценаріїв із залученням генеративного штучного інтелекту вимагає від користувача-розробника чіткого формулювання запитів (промптів), які фактично виконують функцію керування системою.

У процесі проєктування логіки взаємодії з мовною моделлю було обрано підхід, що поєднує використання кількох мов у межах однієї системи. Зокрема, керівні інструкції, опис ролі моделі та правила опрацювання інформації сформульовано англійською мовою, тоді як вхідні дані (тексти повідомлень, документи), а також результати роботи системи зберігаються українською мовою.

Такий вибір має не стільки мовне, скільки функціональне підґрунтя. Сучасні великі мовні моделі були навчені переважно на англійськомовних текстових масивах, що зумовлює їхню вищу чутливість до логічних структур, формалізованих інструкцій і послідовностей дій саме в англійськомовному середовищі. У цьому контексті англійська мова виступає своєрідною «технічною мовою», яка забезпечує точніше виконання складних алгоритмічних вимог.

Практичні спостереження підтверджують, що використання англійської мови в блоці системних інструкцій зменшує ризик неоднозначного тлумачення команд, зокрема під час роботи з правилами класифікації або часовими параметрами. Водночас завдяки здатності моделей до міжмовного перенесення знань англійськомовна логіка ефективно застосовується до україномовного змісту. Це дозволяє зберегти природність мовної взаємодії з користувачем і водночас забезпечити стабільність, передбачуваність і високу точність аналітичних результатів.

У ході дослідження емпірично визначено оптимальний зміст системної інструкції для моделі Gemini:

```
You are an intelligent assistant helping a university professor to track professional achieve.
```

```
Your goal is to analyze the provided IMAGE and the accompanying user CAPTION (if present). This image could be a certificate, diploma, screenshot of a publication, or another document. The caption may contain important details about the event, date, or context that are not visible in the image. Combine information from both sources.
```

```
First, perform Optical Character Recognition (OCR) to extract all relevant text from the image. Pay attention to names, titles, dates, and organization names. Then, based on the extracted text and visual context, analyze the achievement and classify it according to the RATING LIST below.
```

```
RATING LIST (Code - Category Name [Ukrainian]):
```

```
# тут додається повний перелік кодів та категорій досягнень
```

```
INSTRUCTIONS:
```

```
Analyze the input content to understand the type of achievement, date, and context.
```

```
Match the achievement to the most appropriate 'Code' from the Rating List above.
```

```
Extract the Date in format DD-MM-YYYY.
```

```
Reference Date (Today): {{formatDate(now; "DD-MM-YYYY")}}
```

```
Logic for extraction:
```

```
Relative words:
```

```
If the text says "вчора" (yesterday), calculate: [Reference Date] minus 1 day.
```

```
If the text says "позавчора" (day before yesterday), calculate: [Reference Date] minus 2 days.
```

```
If the text says "сьогодні" (today), use [Reference Date].
```

```
Partial dates: If the input contains only Day and Month (e.g., "10 травня"), use the Year from the Reference Date (do not guess previous years).
```

```
No date found: Use [Reference Date].
```

```
Generate a Description in UKRAINIAN language.
```

```
Even if the input is in English, translate the summary to Ukrainian.
```

Format: "[Type]: [Title of article/event/award]".

Extract Link: If a URL is present, put it in 'evidence\_link'. If not, leave empty.

JSON Output: Return strictly a JSON object. No markdown formatting, no intro text.

OUTPUT JSON FORMAT:

```
{ "code": "String (e.g., '2.15')", "category_name": "String (Exact Ukrainian name from the list)", "description": "String (Summary in Ukrainian)", "date": "DD-MM-YYYY", "evidence_link": "String or null" }
```

Наведений вище приклад системного промпту для обробки зображень побудовано з урахуванням сучасних підходів до взаємодії з великими мовними моделями, зокрема принципів поетапного міркування та цілеспрямованого введення контексту. Така організація запиту спрямована на зниження ризику некоректних або довільних інтерпретацій з боку моделі та забезпечення передбачуваної роботи автоматизованого процесу опрацювання даних.

Структурно запит складається з кількох функціонально взаємопов'язаних компонентів, кожен з яких виконує окрему роль у формуванні результату.

Початкова інструкція, що визначає модель як асистента викладача закладу вищої освіти, задає необхідну контекстну рамку для подальшої роботи. Завдяки цьому модель орієнтується на офіційно-діловий стиль мовлення та використовує академічну лексику, релевантну сфері освітньої й наукової діяльності. Таке обмеження контексту зменшує ймовірність появи випадкових або стилістично недоречних відповідей.

Безпосереднє включення переліку рейтингових показників до тексту запиту виконує функцію фіксації семантичних меж завдання. Модель не генерує власні категорії, а здійснює вибір із заздалегідь визначеного набору. Це переводить її діяльність із режиму вільної генерації у режим контрольованої класифікації, що є принципово важливим для подальшого узагальнення та систематизації даних у звітній формі.

Окремий блок інструкцій присвячено роботі з часовими маркерами, які часто мають відносний характер у природному мовленні. У ньому описано правила співвіднесення таких висловів, як «вчора» чи «позавчора», з конкретною датою виконання обробки. Це дає змогу коректно перетворювати усні або текстові повідомлення з невизначеним часом події на чітко зафіксовані календарні дати, придатні для подальшого обліку.

Встановлення жорстких вимог до формату відповіді забезпечує можливість автоматичної взаємодії між мовною моделлю та іншими елементами системи. Структурований вихід у вигляді формалізованого об'єкта дозволяє без додаткової обробки інтегрувати результати в середовище автоматизації та електронні таблиці, що значно спрощує технічну реалізацію та зменшує ризик помилок.

Для перевірки розробленого рішення було проведено експеримент, у якому взяли участь викладачі кафедри музичного мистецтва та хореографії ДЗ «ЛНУ імені Т. Шевченка». Метою було порівняння часових витрат на фіксацію типових досягнень традиційним («ручним») способом та за допомогою чат-бота.

Як тестову вибірку було обрано типи вхідних даних, що покривають найпоширеніші сценарії звітування згідно з чинним «Положенням про рейтингування» (Положення про рейтингування, 2024).

Сценарій А (текстовий): внесення даних про участь у конференції на основі короткої текстової нотатки.

Сценарій Б (візуальний): обробка фотографії сертифіката про підвищення кваліфікації або диплома.

Сценарій В (вебресурс): фіксація наукової статті шляхом передачі посилання на

сторінку журналу.

Сценарій Г (голосовий): внесення даних про проведену профорієнтаційну роботу з голосового повідомлення.

Для кожного сценарію вимірювався час повного циклу операції. У ручному режимі фіксувався час, необхідний для відкриття таблиці, пошуку відповідного пункту в класифікаторі, ручного введення коду, назви, дати, посилання, форматування комірок. В автоматизованому режимі фіксувався час від моменту відправки повідомлення в Telegram до отримання підтвердження про успішний запис у таблицю. Середні значення вимірювань для кожного сценарію (на основі 5 спроб) представлено в таблиці 1.

Табл. 1

Порівняння часових витрат на фіксацію професійного досягнення

Сценарій обробки даних	Ручний режим, хв:с	Автоматизований режим, хв:с	Економія часу, %
Сценарій А (текст)	03:45	00:10	95
Сценарій Б (зображення)	05:20	00:20	94
Сценарій В (посилання)	04:30	00:12	96
Сценарій Г (голос)	03:45	00:14	95
В середньому	04:31	00:14	95

(Джерело: таблицю складено авторами самостійно)

Результати виявилися показовими. При ручному введенні (відкриття таблиці, пошук потрібної графи, набір тексту) фіксація однієї події займала в середньому від 3 до 5 хвилин. Використання бота скоротило цей час до 10-20 секунд.

Особливий інтерес викликають дані щодо голосового введення. У ручному режимі викладачі часто відкладали фіксацію дрібних подій (профорієнтаційні заходи, робота в приймальній комісії) «на потім», що призводило до забування значної частини інформації (ефект втрати даних). Голосовий інтерфейс дозволив фіксувати події в момент їх настання. Точність розпізнавання української мови моделлю Google Gemini 2.5 Flash склала понад 90%, що є прийнятним показником, враховуючи можливість подальшого ручного редагування.

Економічний аналіз також підтвердив ефективність моделі. Використання безкоштовного рівня Make.com (1000 операцій на місяць) повністю покриває потреби одного викладача, який генерує до 100 звітних записів щомісяця. Таким чином, вартість роботи з системою наближається до нуля, що робить її доступною для широкого впровадження без залучення інституційних бюджетів.

Варто підкреслити, що описана система автоматизації це лише окремий приклад, покликаний продемонструвати потенціал концепції. Головна цінність запропонованого підходу полягає в його універсальності. Сучасні платформи оркестрації, такі як Make, надають інтеграцію з сотнями сервісів, дозволяючи кожному науковцю конструювати унікальні рішення під власні специфічні завдання.

За аналогічним принципом користувач-розробник може автоматизувати й інші рутинні аспекти академічної діяльності. Серед перспективних сценаріїв можна виділити: моніторинг грантових можливостей, автоматичну генерацію квізів для студентів на основі текстів лекцій або створення персоналізованих нагадувань про дедлайни, сортування вхідних запитів від абітурієнтів та формування чернеток відповідей. Таким чином, технологія дозволяє перейти від вирішення однієї проблеми до створення персональної екосистеми цифрових асистентів.

Для відтворення експерименту, аналізу промптів та адаптації рішення під власні потреби, читачі можуть скористатися розробленим сценарієм, який наразі є у вільному доступі на платформі Make (AI-асистент викладача, 2025). Для роботи сценарію необхідно налаштувати власні ключі доступу до сервісів Telegram та Google. З огляду на

динамічність регуляторного середовища та обговорення можливих обмежень у використанні месенджера Telegram в Україні, важливо зазначити, що запропонована архітектура має зберігати свою працездатність і при заміні інтерфейсу взаємодії. Платформа Make.com забезпечує інтеграцію з альтернативними каналами комунікації, такими як месенджери Slack, Discord, сервісом електронної пошти Gmail, або Google Forms.

**Обмеження дослідження.** Незважаючи на високу ефективність запропонованої архітектури на основі платформи Make.com та мовної моделі Google Gemini у зменшенні адміністративного навантаження викладача, практичне впровадження підходу «користувач-розробник» супроводжується низкою технічних і нормативних викликів.

Економічна доцільність запропонованого рішення значною мірою пов'язана з можливістю використання безкоштовного тарифного плану платформи Make.com, який передбачає обмеження у 1000 операцій на місяць. У термінах платформи кожне виконання окремого модуля розглядається як самостійна операція. Тому створення складних розгалужених автоматизацій з великою кількістю модулів, або активне тестування системи чи її помилкові запуски – може перевищити пропускну здатність безкоштовного тарифу. Це вимагає від користувача-розробника свідомого та ощадливого використання ресурсу або ж переходу на розширені тарифні плани.

Використання сервісів штучного інтелекту неминує порушує питання захисту даних та відповідності етичним нормам. Відповідно до умов надання послуг, інформація, що передається до безоплатних версій моделей, може використовуватися для вдосконалення алгоритмів та, за необхідності, переглядатися фахівцями з контролю якості. З цієї причини система не повинна застосовуватися для опрацювання персональних даних студентів, внутрішньої службової інформації або матеріалів, що становлять об'єкти інтелектуальної власності до моменту їх офіційного оприлюднення.

Використання месенджера Telegram як інтерфейсу взаємодії з автоматизованою системою також має свої обмеження. Функціонування чат-ботів передбачає серверну обробку повідомлень, що унеможливує застосування наскрізного шифрування в повному обсязі. Хоча така архітектура є типовою для більшості бот-платформ, вона вимагає від викладача підвищеної уваги до характеру переданої інформації. Її доцільно використовувати для фіксації фактів професійної діяльності, дат, назв заходів або публічних подій, уникаючи передачі змісту приватного листування чи ідентифікаційних даних третіх осіб.

Запобігання таким ризикам можливе шляхом дотримання принципів так званої «гігієни даних». Зокрема, перед передачею будь-яких матеріалів до автоматизованої системи користувач має переконатися, що вони не містять персональних ідентифікаційних даних студентів або колег, а також не належать до категорії службової документації закладу.

## **ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ**

Проведене дослідження підтвердило високу ефективність запропонованої моделі персональної автоматизованої системи обліку професійної активності викладача як засобу мінімізації адміністративного навантаження. Розроблений прототип, у якому No-Code платформа Make.com використовується як середовище проектування сценаріїв, а мультимодальна модель Google Gemini 2.5 Flash – як зовнішній когнітивний агент, засвідчує скорочення часових витрат на фіксацію досягнень на 95% (з 4:31 хв до 0:14 с), що вивільняє значні ресурси для наукової, педагогічної та творчої діяльності.

Запропонована архітектура з динамічним маршрутизатором та оптимізованою системою промптів забезпечує точну (понад 90%) класифікацію різномірних даних – від

текстових описів і голосових повідомлень до зображень сертифікатів та вебресурсів – з автоматичним формуванням структурованих звітів у Google Sheets.

Економічна доцільність та доступність рішення (безкоштовні тарифи покривають до 100 записів/місяць) робить його придатним для індивідуального впровадження без залучення інституційних бюджетів. Концепція «користувача-розробника» як нової ролі викладача в цифрову епоху отримала емпіричне обґрунтування: викладачі без ІТ-підготовки здатні створювати персоналізовані цифрові екосистеми, адаптовані до специфіки педагогічної та мистецької діяльності. Це сприяє переходу від пасивного споживання ІКТ до активного конструювання рішень, що відповідає сучасним викликам цифрової трансформації вищої освіти.

Перспективи подальших досліджень охоплюють масштабування системи на рівні факультету/університету, інтеграцію з корпоративними системами, підвищення безпеки даних, створення автоматизованих рішень під інші потреби освітян. Запропонований підхід має потенціал для широкого впровадження в українських закладах освіти як елемент стратегії зменшення бюрократизації академічного середовища.

### **Конфлікт інтересів**

Автори заявляють про відсутність будь-якого конфлікту інтересів, зокрема професійних, наукових або особистих обставин, які могли б вплинути на об'єктивність подання, рецензування чи ухвалення редакційних рішень щодо цієї статті. Дослідження є ініціативною науковою роботою, яка не отримувала фінансової підтримки, грантів або винагород від сторонніх комерційних чи некомерційних організацій, що могло б створити потенційний конфлікт інтересів. Результати та висновки, викладені в роботі, ґрунтуються виключно на емпіричних даних, отриманих під час апробації системи.

### **Декларування використання інструментів штучного інтелекту**

Під час підготовки дослідження автори використовували інструмент на основі штучного інтелекту Gemini 3 для технічного опрацювання тексту, а саме для перевірки коректності англomовного перекладу анотації та ключових слів. Після застосування зазначеного інструменту весь матеріал було додатково переглянуто, критично оцінено та відредаговано авторами для забезпечення точності термінології та збереження авторського стилю. Слід зазначити, що використання мультимодальної моделі Google Gemini 2.5 Flash, описане в основному тексті статті, є безпосереднім об'єктом наукового дослідження та ключовим компонентом розробленої архітектури автоматизації, а не засобом автоматичного генерування тексту самого рукопису. Авторі зберігають повний контроль над змістом публікації та несуть відповідальність за її достовірність, правдивість та оригінальність відповідно до політики редакції.

### **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

- Woelert, P., Chesters, J., Martinussen, M., & Gannaway, J. (2025). Administrative burden in Australian universities: Insights into dimensions and drivers from a nationwide survey. *Science and Public Policy*. <https://doi.org/10.1093/scipol/scaf029>
- Schiller, J., & LeMire, S. (2023). A survey of research administrators: Identifying administrative burden in post-award federal research grant management. *The Journal of Research Administration*, 55(3), 9-28. <https://files.eric.ed.gov/fulltext/EJ1412033.pdf>
- Moynihan, D., Herd, P., & Harvey, H. (2014). Administrative burden: Learning, psychological, and compliance costs in citizen-state interactions. *Journal of Public Administration Research and Theory*, 25(1), 43-69. <https://doi.org/10.1093/jopart/muu009>

- Davidovitch, N., & Cohen, E. (2024). Administrative roles in academia – potential clash with research output and teaching quality? *Cogent Education*, 11(1). <https://doi.org/10.1080/2331186x.2024.2357914>
- Bhardwaj, V., & Kumar, M. (2025). Transforming higher education with robotic process automation: Enhancing efficiency, innovation, and student-centered learning. *Discover Sustainability*, 6(1). <https://doi.org/10.1007/s43621-025-01198-6>
- Pinho, D., Aguiar, A., & Amaral, V. (2022). What about the usability in low-code platforms? A systematic literature review. *Journal of Computer Languages*, 101185. <https://doi.org/10.1016/j.cola.2022.101185>
- McHugh, S., Carroll, N., & Connolly, C. (2023). Low-Code and No-Code in secondary education – empowering teachers to embed citizen development in schools. *Computers in the Schools*, 1-26. <https://doi.org/10.1080/07380569.2023.2256729>
- Sodano, J., & DeFranco, J.(2025). Citizen development, Low-Code/No-Code platforms, and the evolution of generative AI in software development. *Computer*, 58(5), 101-104. <https://doi.org/10.1109/mc.2025.3547073>
- Умрик, М., Морзе, Н., & Смирнова-Трибульська, Є. (2025). Розвиток компетентностей освітян у галузі використання штучного інтелекту в цифровому суспільстві. *Електронне наукове фахове видання “Відкрите освітнє е-середовище сучасного університету”*, 18, 159-173. <https://doi.org/10.28925/2414-0325.2025.1813>
- Гужва, В. (2024). Цифрова трансформація університетів: Інтелектуальні агенти на основі low-code технологій. *Економіка та суспільство*, 70. <https://doi.org/10.32782/2524-0072/2024-70-4>
- Івасенко, О. (2025). Соціально-економічна ефективність використання low-code платформ у цифровому маркетингу закладів вищої освіти. *Економіка і регіон*, 3(98), 225-232. [https://doi.org/10.26906/eir.2025.3\(98\).4035](https://doi.org/10.26906/eir.2025.3(98).4035)
- Положення про рейтингування науково-педагогічних працівників ДЗ «Луганський національний університет імені Тараса Шевченка»*. (2024). Полтава. [https://luguniv.edu.ua/wp-content/uploads/2024/12/polozh\\_pro\\_reitynhuvannia\\_2024.pdf](https://luguniv.edu.ua/wp-content/uploads/2024/12/polozh_pro_reitynhuvannia_2024.pdf)
- AI-асистент викладача: Telegram + Gemini – Make.com Automation Scenario*. (2025). Make.com. <https://eu1.make.com/public/shared-scenario/wrKVPa3XUz5/ai-asistent-vikladacha-telegram-gemini>

Надходження статті до видання 20.02.2026 р.

Прийняття статті до друку після рецензування 15.04.2026 р.

Дата публікації 24.04.2026 р.

## AUTOMATION OF ACADEMIC STAFF REPORTING USING ARTIFICIAL INTELLIGENCE AND NO-CODE PLATFORM

### **Tetiana Ostretsova**

Senior Collaborative Pianist of the Department of Musical Art and Choreography at the Faculty of Arts  
Luhansk Taras Shevchenko National University, Poltava, Ukraine

[t.axenova@gmail.com](mailto:t.axenova@gmail.com)

ORCID: [0009-0001-9584-0365](https://orcid.org/0009-0001-9584-0365)

### **Olena Ponomarenko**

Collaborative Pianist of the Department of Musical Art and Choreography at the Faculty of Arts  
Luhansk Taras Shevchenko National University, Poltava, Ukraine

[elenapon91@gmail.com](mailto:elenapon91@gmail.com)

ORCID: [0009-0004-9053-2701](https://orcid.org/0009-0004-9053-2701)

**Anna Yezerka**

Collaborative Pianist of the Department of Musical Art and Choreography at the Faculty of Arts  
Luhansk Taras Shevchenko National University, Poltava, Ukraine

[ezerskaorcid@gmail.com](mailto:ezerskaorcid@gmail.com)

ORCID: [0009-0000-4203-7571](https://orcid.org/0009-0000-4203-7571)

**Abstract.** In modern higher education institutions, the issue of excessive administrative burden has become critical. This often leads to the "digital paradox" phenomenon, where the implementation of centralized information systems, instead of streamlining operations, merely increases the time spent on reporting. This study proposes a solution to this problem through the development and empirical verification of a personal automated system for tracking faculty professional activity. The methodological framework for this work is the Design Science Research (DSR) approach and the "Citizen Developer" concept, which empowers educators without specialized technical training to create their own adaptive digital tools. The proposed architecture integrates the Telegram messenger, the Make.com platform as a scenario orchestration environment, and the Google Gemini 2.5 Flash multimodal artificial intelligence model. The developed algorithm facilitates complex routing for heterogeneous input data, including text descriptions, voice messages, web links (e.g., to scientific publications or news), and images (e.g., certificates or event posters). The system performs semantic content analysis, extracts key dates, generates descriptions in Ukrainian, and automatically logs entries into Google Sheets with linked digital copies of the relevant documents. Experimental validation conducted at the Faculty of Arts of Luhansk Taras Shevchenko National University (Poltava) demonstrated a reduction in the time required to record a single achievement from 4 minutes 31 seconds to 14 seconds, representing a 95% time saving. The paper analyzes the technical aspects of configuring system prompts to achieve a classification accuracy exceeding 90% and addresses information security concerns. The results demonstrate that the strategic use of accessible No-Code solutions enables the transformation of routine bureaucratic procedures into efficient processes, thereby freeing up faculty resources for scientific and creative endeavors.

**Keywords:** reporting automation; Citizen Developer; Low-Code/No-Code; artificial intelligence; Google Gemini; Make.com

**REFERENCES (TRANSLATED AND TRANSLITERATED)**

- Woelert, P., Chesters, J., Martinussen, M., & Gannaway, J. (2025). Administrative burden in Australian universities: Insights into dimensions and drivers from a nationwide survey. *Science and Public Policy*. <https://doi.org/10.1093/scipol/scaf029>
- Schiller, J., & LeMire, S. (2023). A survey of research administrators: Identifying administrative burden in post-award federal research grant management. *The Journal of Research Administration*, 55(3), 9-28. <https://files.eric.ed.gov/fulltext/EJ1412033.pdf>
- Moynihan, D., Herd, P., & Harvey, H. (2014). Administrative burden: Learning, psychological, and compliance costs in citizen-state interactions. *Journal of Public Administration Research and Theory*, 25(1), 43-69. <https://doi.org/10.1093/jopart/muu009>
- Davidovitch, N., & Cohen, E. (2024). Administrative roles in academia – potential clash with research output and teaching quality? *Cogent Education*, 11(1). <https://doi.org/10.1080/2331186x.2024.2357914>
- Bhardwaj, V., & Kumar, M. (2025). Transforming higher education with robotic process automation: Enhancing efficiency, innovation, and student-centered learning. *Discover Sustainability*, 6(1). <https://doi.org/10.1007/s43621-025-01198-6>
- Pinho, D., Aguiar, A., & Amaral, V. (2022). What about the usability in low-code platforms? A systematic literature review. *Journal of Computer Languages*, 101185. <https://doi.org/10.1016/j.cola.2022.101185>
- McHugh, S., Carroll, N., & Connolly, C. (2023). Low-Code and No-Code in secondary education – empowering teachers to embed citizen development in schools. *Computers in the Schools*, 1-26. <https://doi.org/10.1080/07380569.2023.2256729>

- Sodano, J., & DeFranco, J.(2025). Citizen development, Low-Code/No-Code platforms, and the evolution of generative AI in software development. *Computer*, 58(5), 101-104. <https://doi.org/10.1109/mc.2025.3547073>
- Umryk, M., Morze, N., & Smirnova-Trybulska, E. (2025). Development of artificial intelligence competences for educators in the digital society. *Electronic Scientific Professional Journal "Open Educational E-Environment of Modern University"*, 18, 159-173. <https://doi.org/10.28925/2414-0325.2025.1813> (in Ukrainian).
- Huzhva, V. (2024). Digital transformation of universities: Intelligent agents based on low-code technologies. *Economy and Society*, 70. <https://doi.org/10.32782/2524-0072/2024-70-4> (in Ukrainian).
- Ivasenko, O. (2025). Socio-economic efficiency of using low-code platforms in digital marketing of higher education institutions. *Economics and region*, 3(98), 225-232. [https://doi.org/10.26906/eir.2025.3\(98\).4035](https://doi.org/10.26906/eir.2025.3(98).4035) (in Ukrainian).
- Regulations on the Performance Evaluation of Academic and Teaching Staff of the Luhansk Taras Shevchenko National University*. (2024). Poltava. [https://luguniv.edu.ua/wp-content/uploads/2024/12/polozh\\_pro\\_reitynhuvannia\\_2024.pdf](https://luguniv.edu.ua/wp-content/uploads/2024/12/polozh_pro_reitynhuvannia_2024.pdf) (in Ukrainian).
- AI Assistant for Educators: Telegram – Gemini Integration via Make.com Automation Scenario* (2025). Make.com. <https://eu1.make.com/public/shared-scenario/wrKVPa3XUz5/ai-asistent-vikladacha-telegram-gemini> (in Ukrainian).

