

Ілуца А.С.<https://orcid.org/0009-0007-3445-8481>

Національний університет «Одеська політехніка»

ВЕБ-ОРІЄНТОВАНІ ПРОГРАМНІ ЗАСОБИ ІНТЕГРАЦІЇ PaaS I SaaS ДЛЯ ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ ДОСЛІДНИЦЬКОЇ ТА ОСВІТНЬОЇ ДІЯЛЬНОСТІ

Метою роботи є розробка веб-орієнтованих програмних засобів, що поєднують сервіси PaaS і SaaS та забезпечують підвищення ефективності освітньої діяльності та міждисциплінарних досліджень. Розглянуто проблему організації хмарних обчислень в умовах цифрової трансформації освіти та науки. Показано, що використання локальних обчислювальних ресурсів і розмежування між середовищами програмної розробки та готовими інструментальними засобами у вигляді GUI-інтерфейсів знижує гнучкість, масштабованість і ефективність дослідницького та освітнього процесів. Проведений аналіз існуючих рішень демонструє, що вони не забезпечують комплексного поєднання можливостей розробки програмного коду, створення інтерфейсів та організації спільної роботи в межах єдиного середовища. Розроблено веб-орієнтовані програмні засоби, що забезпечують: одночасну роботу з програмним кодом на мовах програмування Python і JavaScript та з GUI-інтерфейсами; виконання обчислень на сервері; зберігання результатів та їх візуалізацію незалежно від мови програмування; можливість інтеграції сторонніх сенсорів для проведення експериментів та накопичення і повторного використання даних з них; організацію спільної взаємодії користувачів. Реалізовано механізм взаємодії між інтерфейсом і програмною логікою, за рахунок розробленого конструктора інтерфейсів та транслятора коду, що дозволяє змінювати умови виконання досліджень без редагування вихідного коду. Додатково забезпечується підвищення відтворюваності результатів досліджень та адаптивність до різних предметних областей завдяки модульності й гнучкому налаштуванню процесу досліджень, що досягається за рахунок збереження вхідних даних, параметрів виконання та результатів досліджень. Запропоноване рішення забезпечує підвищення продуктивності дослідницької та освітньої діяльності за рахунок інтеграції процесів розробки, виконання обчислень і використання прикладних інструментів, у вигляді GUI-інтерфейсів, в межах єдиного веб-орієнтованого середовища.

Ключові слова: веб-платформа, програмні засоби, дистанційні дослідження, дистанційна освіта, цифрова освіта, PaaS, SaaS, хмарні технології, хмарні обчислення.

Постановка проблеми. Сучасний розвиток освіти й науки відбувається в умовах глибокої цифрової трансформації, що змінює підходи до організації навчального процесу та дослідницької діяльності. Традиційна практика використання локальних обчислювальних ресурсів разом з встановленим програмним забезпеченням дедалі частіше виявляється недостатньою через обмеженість апаратних потужностей, складність технічного супроводу та відсутність гнучкого масштабування. За таких умов можна дедалі частіше спостерігати перехід до хмарних підходів організації дослідницького та освітнього процесів, які забезпечують доступність, масштабованість і адаптивність інструментів проведення досліджень незалежно від місця перебування користувача.

Особливо гостро ця проблема постає у сфері досліджень, пов'язаних з аналізом великих масивів експериментальних даних та моделюванням складних процесів. Наприклад, у нейрофізіологічних і психофізіологічних дослідженнях активно застосовуються технології стеження за рухами очей (eye-tracking) [1–5], яка використовується для вивчення когнітивних процесів [6], наприклад моніторингу стану студентів під час навчання [7]. Такі експерименти супроводжуються накопиченням значних обсягів структурованих і неструктурованих даних, що потребує ефективних механізмів їх зберігання, обробки та повторного використання. Водночас виникає потреба забезпечити спільну роботу дослідників, відтворюваність результатів і гнучке налаштування процесу досліджень без надмірної залежності від конкретної локальної інфраструктури.



Для вирішення подібних завдань доцільне використання хмарних технологій, оскільки такі технології забезпечують централізований доступ до серверів, баз даних, сховищ і програмних інструментів через мережу Інтернет [8–11]. Проте на практиці часто спостерігається розрив між середовищами, призначеними для виконання експериментів, де користувач взаємодіє виключно з готовим інтерфейсом, і середовищами розробки інструментальних засобів обробки даних на рівні програмного коду, що ускладнює формування цілісного науково-освітнього простору. Таким чином, постає завдання створення інтегрованого хмарного рішення, яке б забезпечувало повний цикл роботи з експериментальними даними – від їх отримання до аналітичної обробки та інтерпретації результатів.

Аналіз останніх досліджень і публікацій. Хмарні обчислення розглядаються як багатоврівнева організація доступу до обчислювальних ресурсів і програмних сервісів, що реалізується через різні підходи до взаємодії з користувачем: програмне забезпечення як сервіс (SaaS), платформа як сервіс (PaaS), інфраструктура як сервіс (IaaS) та функція як сервіс (FaaS) [12–14]. Кожен із цих підходів орієнтований на певний рівень абстракції – від використання готових програмних продуктів до керування обчислювальною інфраструктурою та виконання окремих функцій. У контексті науково-освітньої діяльності особливу увагу привертає поєднання можливостей програмної розробки інструментальних засобів з використанням готових інструментів керування експериментами (інтерфейсів), що дозволяє поєднати програмну гнучкість із зручністю інтерфейсної взаємодії.

У сфері хмарних обчислень для освітньої та науково-дослідної діяльності активно використовуються такі програмні рішення, як Project Jupyter [15], Google Colab [16], OnlineGDB [17] та MATLAB Online [18]. Незважаючи на широку популярність, зазначені засоби мають низку концептуальних та функціональних обмежень, що ускладнюють їх використання в комплексних міждисциплінарних дослідженнях і в організації повноцінного освітнього процесу.

Project Jupyter є одним із найбільш поширених інструментів для інтерактивної роботи з даними. Він забезпечує створення проєктів (середовищ), у яких поєднуються програмний код та текстові пояснення. Середовище не надає можливостей використання декількох мов програмування та орієнтоване тільки на мову програмування

Python. Крім того, виконання обчислень залежить від локальних ресурсів або потребує окремої конфігурації віддалених серверів. Засоби створення повноцінних графічних інтерфейсів користувача відсутні, а інтеграція з апаратними сенсорами та зовнішнім обладнанням не передбачена. Також не передбачено зберігання результатів досліджень.

Google Colab є веб-орієнтованою реалізацією ідей Jupyter із використанням хмарної інфраструктури Google. Перевагою є доступ до обчислювальних ресурсів та можливість колективної роботи (шеринг проєктів). Разом із тим, середовище повністю орієнтоване на Python. Сесії мають часові обмеження, що створює ризик втрати даних при тривалих експериментах. Функції зберігання результатів досліджень, інструменти створення інтерактивних графічних інтерфейсів та інтеграції з апаратними засобами для проведення експериментів відсутні.

OnlineGDB є багатомовним онлайн-компілятором, який підтримує C, C++, Java, Python, JavaScript та інші мови програмування. Його основна перевага – швидке тестування окремих фрагментів коду без встановлення локального програмного забезпечення. Проте функціональність обмежується саме компіляцією та запуском програм. Засоби колективної роботи, побудови GUI, організації експериментів або підключення апаратних сенсорів не реалізовані. Таким чином, цей інструмент орієнтований переважно на навчальні завдання початкового рівня.

MATLAB Online надає доступ до обчислювальних можливостей MATLAB через веб-браузер. Він ефективний для інженерних розрахунків, обробки сигналів і побудови графіків. Водночас підтримується виключно мова MATLAB, що є недоліком. Можливості створення повноцінних користувацьких графічних інтерфейсів відсутні порівняно з локальною (offline) версією. Засоби інтеграції із зовнішнім обладнанням у веб-версії не передбачені. Колективна взаємодія та зберігання результатів досліджень відсутні.

Порівняльний аналіз показує, що наявні програмні засоби:

- орієнтовані переважно на одну мову програмування;
- не підтримують створення складних користувацьких GUI;
- не передбачають інтеграції з апаратними сенсорами та експериментальним обладнанням;
- не поєднують сервіси PaaS і SaaS для комплексної організації освітньої та дослідницької діяльності у межах єдиного середовища;

– не надають можливості зберігати дані експериментів та результати досліджень.

У результаті жоден із розглянутих засобів повною мірою не забезпечує підтримки міждисциплінарних досліджень, у яких одночасно необхідні:

- робота з програмним кодом кількома мовами;
- створення та використання готових графічних інтерфейсів;
- інтеграція зовнішнього експериментального обладнання;
- централізоване зберігання та обробка даних;
- підтримка взаємодії між дослідниками.

У таблиці 1 наведено порівняння ключових можливостей існуючих рішень організації хмарних обчислень та потреб для підвищення продуктивності досліджень та освітнього процесу.

У результаті аналізу можемо зазначити, що сучасні освітні та наукові процеси потребують програмних засобів, що одночасно надають PaaS та SaaS сервіси для організації дослідницького і освітнього процесів та дозволяють забезпечити:

- роботу з багатомовними проектами;
- створення та використання готових прикладних інтерфейсів;
- інтеграцію з експериментальним обладнанням;
- централізоване зберігання та обробку даних;
- підтримку дослідницької та навчальної діяльності в єдиному інформаційному середовищі.

Постановка завдання. Таким чином, метою роботи є розробка веб-орієнтованих програмних засобів, що поєднують сервіси PaaS і SaaS та підвищують продуктивність дослідницького та освітнього процесів.

Виклад основного матеріалу. Розроблено веб-орієнтовані програмний комплекс організації хмарних обчислень IONDS (*In On Data Science*) [19] для підвищення ефективності дослідницького та освітнього процесів, що реалізується завдяки поєднанню сервісів PaaS (платформа як сервіс) та SaaS (програмне забезпечення як сервіс). Такий підхід забезпечує дослідникам можливість одночасної роботи з програмним кодом, через вбудований редактор скрипт-коду, та з інтерфей-

сом вже реалізованих інструментальних засобів, який може бути створений завдяки інтегрованому конструктору. Використовуючи редактор коду та конструктор інтерфейсів, користувачі можуть створювати нові або масштабувати та редагувати вже створені інструментальні засоби обробки даних, запускати обчислення на сервері (у хмарі) та отримувати результати безпосередньо у браузері. Розроблені програмні засоби являють собою програмний комплекс, що складається з багатьох модулів і вузлів, які взаємодіють між собою [20].

Однією з ключових складових розроблених програмних засобів є конструктор інтерфейсів, що працює разом з транслятором коду. Їх основне призначення - реалізація механізму декларативного опису параметрів, який дозволяє користувачам без глибоких технічних знань змінювати параметри досліджень без необхідності прямого редагування скрипт-коду інструментальних засобів, а також візуалізувати результати досліджень без залежності від мови програмування [21]. Завдяки цьому забезпечується високий рівень абстракції (доступності), що є критично важливим у міждисциплінарних дослідженнях, де користувачами системи можуть бути представники різних галузей: інженери, студенти або науковці.

Також, розроблений програмний комплекс має можливість підключення до програмних засобів сторонніх сенсорів (датчиків), наприклад камер або біометричних пристроїв, для автоматичного збору даних. Сторонні апаратні засоби (сенсори) під'єднуються безпосередньо до серверу (Рис. 1) завдяки спеціальному програмному API. Він дає змогу передавати дані з будь-якого зовнішнього застосунку безпосередньо у сховище IONDS. Отримана від сенсорів інформація передається на сервер, де вона зберігається в базі даних для подальшого використання у дослідженнях. У процесі передачі дані автоматично структуруються за заданими категоріями та зберігають нативний формат (масиви, об'єкти), що значно полегшує їх подальше використання у проектах. Варто зазначити, що результати експериментів зберігаються для кожного користувача окремо за для забез-

Таблиця 1

Порівняння можливостей існуючих рішень для організації хмарних обчислень та сучасних потреб

	Project Jupyter	Google Colab	OnlineGDB	MATLAB Online	Необхідне рішення
Підтримка кількох мов програмування	Відсутня	Відсутня	Присутня	Відсутня	Присутня
Колективна робота	Відсутня	Присутня	Відсутня	Відсутня	Присутня
Створення та використання GUI-інтерфейсів	Відсутнє	Відсутнє	Відсутнє	Відсутнє	Присутнє
Зберігання даних та результатів	Відсутнє	Відсутнє	Відсутнє	Відсутнє	Присутнє
Інтеграція апаратних сенсорів	Відсутня	Відсутня	Відсутня	Відсутня	Присутня

печення конфіденційності. Для цього необхідно передати, у якості значення поля authors, перелік UID користувачів, для яких результати можуть бути доступними, а також пройти авторизацію, використовуючи спеціальний токен, який видається адміністратором платформи.

Розроблені програмні засоби складаються з семи основних сторінок, які пов'язані між собою (Рис. 2) та дозволяють комплексно проводити дослідження.

Робота з програмними засобами починається з сторінки входу (Рис. 3), на якій користувачі можуть отримати доступ до функцій комплексу, ввівши своє ім'я користувача та пароль, або обравши авторизацію через Google-акаунт.

Після авторизації, користувач потрапляє на головну сторінку комплексу (Рис. 4), яка виконує роль інформаційної панелі, що консолідує всі ключові дані для користувача: статистика виконання проєктів, остання активність акаунту, останні

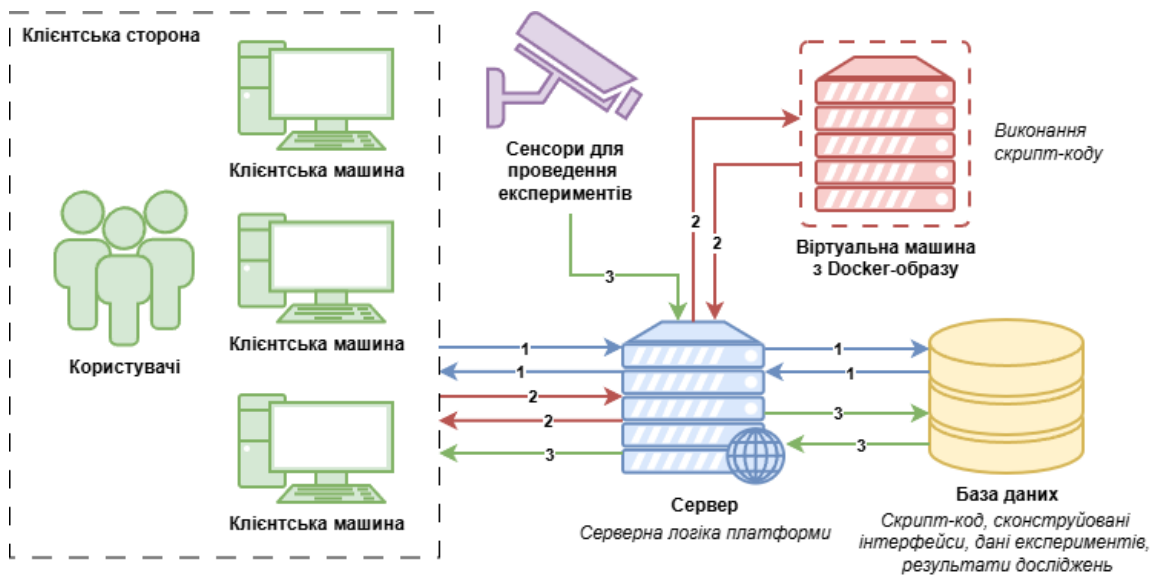


Рис. 1. Схема під'єднання сенсорів до серверної частини комплексу



Рис. 2. Зв'язок основних сторінок комплексу

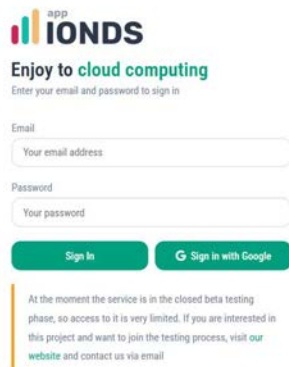


Рис. 3. Сторінка авторизації

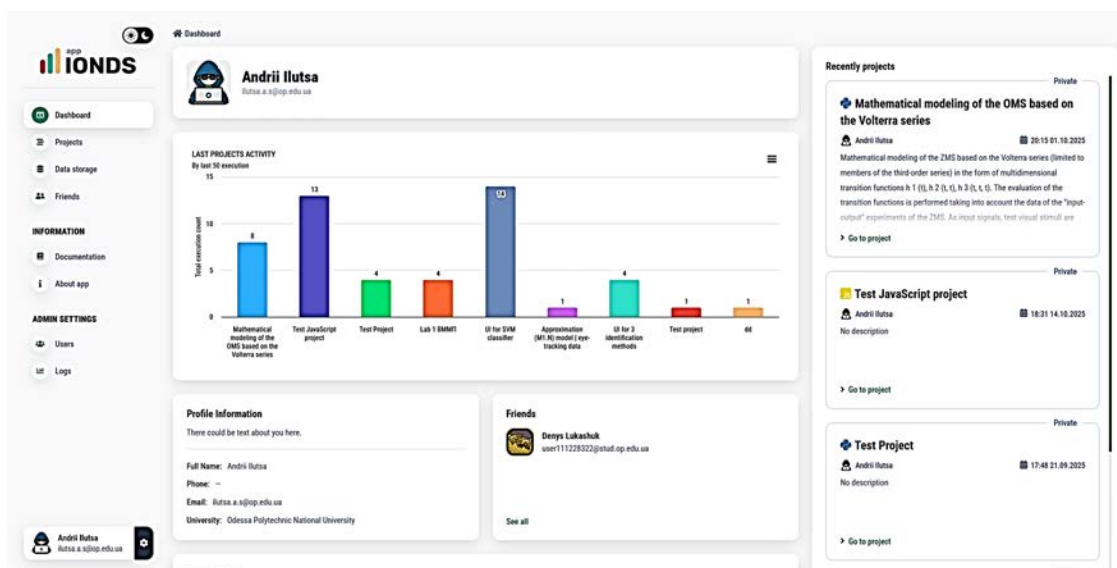


Рис. 4. Головна сторінка комплексу

використані проекти, перелік обраних акаунтів та особиста інформація профілю.

Також, на головній та всіх інших сторінках користувач має доступ до головного меню комплексу (Рис. 5), яке містить: логотип, перемикач режиму відображення (світлий, темний), секцію користувача (user-box) та перелік посилань на ключові сторінки, які дають змогу швидко отримувати доступ до ключових сторінок комплексу, а саме: головна сторінка, перелік проектів, сховище даних експериментів, сторінка обраних акаунтів, документація, сайт комплексу.

Після головної сторінки користувач може перейти на сторінку проектів (Рис. 6), яка надає доступ до переліку публічних проектів, особистих проектів користувача та проектів, якими поділилися з користувачем інші користувачі комплексу. Всі ці проекти згруповані по відповідним вкладкам, між

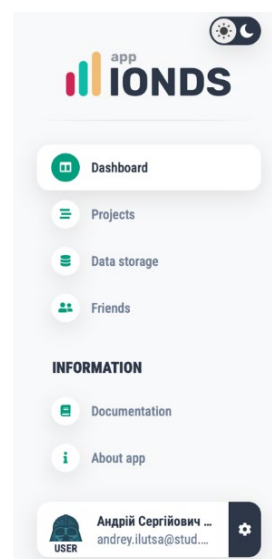


Рис. 5. Головне меню комплексу

якими користувач може перемикається.

Картка проекту, яка відображається на сторінці, містить короткий опис: назву, дату останнього редагування, автора та короткий опис проекту, якщо його вказано при створенні. На цій сторінці користувачі можуть створювати нові публічні або приватні проекти.

Коли користувач обирає та відкриває певний проект, він потрапляє на його сторінку. На прикладі проекту пов'язаного з оцінкою психофізіологічного стану людини з використанням платформи хмарних обчислень [22] можемо бачити, як виглядає сторінка проекту з створеним інтер-

фейсом. На ній ми можемо обирати файл з результатами експериментів або звернутися до сховища даних експериментів, та взаємодіяти з створеним спеціалізованим інтерфейсом для управління параметрами досліджень, де є можливість керувати параметрами: $\$Identification_Method\$, \$N\$, \$L\$, \$all_amplitudes\$, \$base_amplitudes\$$ (Рис. 7).

Функціональність сторінки проекту залежить від того, чи є користувач автором, чи просто переглядачем. Якщо користувач є автором, він може: редагувати код, конструювати або використовувати інтерфейс для налаштування параметрів досліджень, обирати результати експеримен-

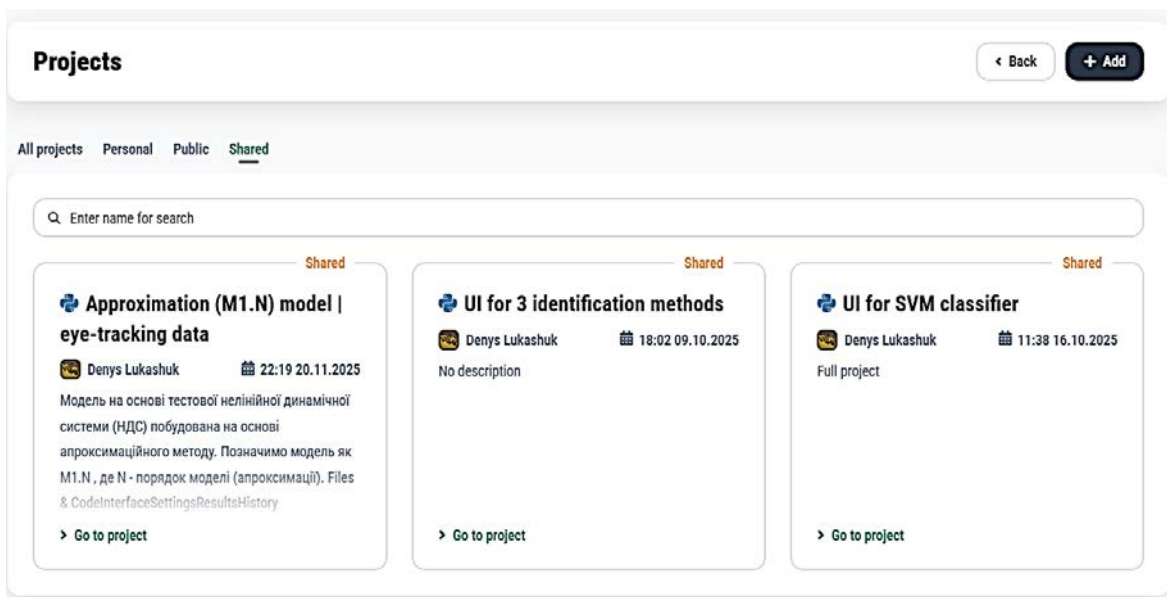


Рис. 6. Сторінка переліку проектів

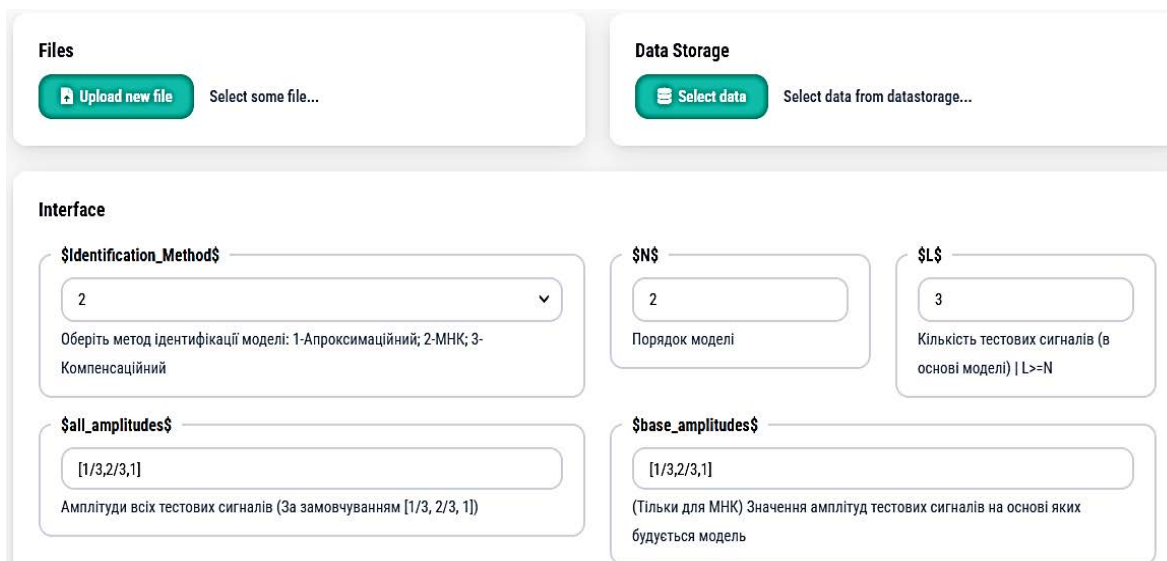


Рис. 7. Сторінка проекту з створеним інтерфейсом

тів з спеціального сховища або завантаживши таблиці, запускати обчислення, переглядати результати досліджень у текстовому та графічному форматах незалежно від мови програмування завдяки розробленому транслятору коду та мові шаблонів перетворень [21]. Якщо ж проєкт належить іншому досліднику і є публічним, користувач не може його редагувати, але все одно має змогу запускати код, змінювати параметри досліджень (якщо для проєкта створено інтерфейс) та переглядати результати у текстовому та графічному форматах. Варто зазначити, що всі результати досліджень, для кожного користувача, зберігаються у спеціальній вкладці на сторінці проєкту, що дає змогу користувачу у будь який момент повернутися до необхідних результатів для аналізу без потреби виконувати обчислення знову. Таким чином, сторінка проєкта стає основним робочим простором, де дослідження або розробляються розробником, або використовуються дослідником.

Також, приватними проєктами можна ділитися с іншими користувачами, що дозволить проводити спільні дослідження. У таких користувачів ці проєкти відображаються у окремій вкладці, та мають трохи обмежені права управління, на відміну від приватних: немає можливості змінювати опис проєкту та видаляти його.

Окремо від сторінки проєкту, користувач може, через головне меню, потрапити на сторінку сховища даних експериментів (Рис. 8), яка забезпечує централізоване та структуроване накопичення та зберігання цих даних.

Основним завданням сховища є зберігання та організація результатів експериментів у єдиній базі, з якої дані можуть бути повторно використані у різних дослідженнях в межах проєктів комплексу. Данні експериментів зберігаються за принципом категорій (директорій), у межах яких групуються окремі експерименти (Рис. 8). Такий

підхід дає змогу підтримувати логічну організацію великої кількості результатів, завдяки чому дослідник отримує не лише сховище, а й інструмент управління результатами експериментів, що значно спрощує та підвищує продуктивність процесу досліджень. Також, додавати результати експериментів у сховище можна як через спеціальне API, так і через вбудований інтерфейс, заповнюючи всі вище наведені поля у ручному режимі.

Сторінка з налаштуваннями акаунту (сторінка користувача) (Рис. 9), на яку також можна потрапити через головне меню, використовується для ознайомлення та редагування особистої інформації користувача, а саме: фото користувача (аватар), ім'я, телефон, email, заклад освіти, інформація про користувача.

Також, знаходячись на цій сторінці користувач має змогу вивантажувати, або завантажувати файл з результатами досліджень. Для того, щоб вивантажити результати своїх досліджень, користувач повинен натиснути кнопку «Download projects history», після чого на його комп'ютер буде завантажено JSON-файл, у котрому структуровано зберігаються усі результати досліджень. Також, цей файл можна передати іншому користувачу для завантаження (для цього треба натиснути кнопку «Upload history file») та отримання доступу до відповідних результатів досліджень.

Структура даних розроблених програмних засобів організована таким чином, щоб забезпечити повний цикл роботи: збереження результатів експериментів, виконання досліджень на їх основі та фіксацію отриманих результатів. Вона складається з чотирьох основних колекцій: users, projects, data та logs. Кожна з них відповідає за окремий функціональний рівень.

Колекція users забезпечує управління обліковими записами та має наступну структуру:

- id – унікальний ідентифікатор користувача;
- created – дата створення запису;

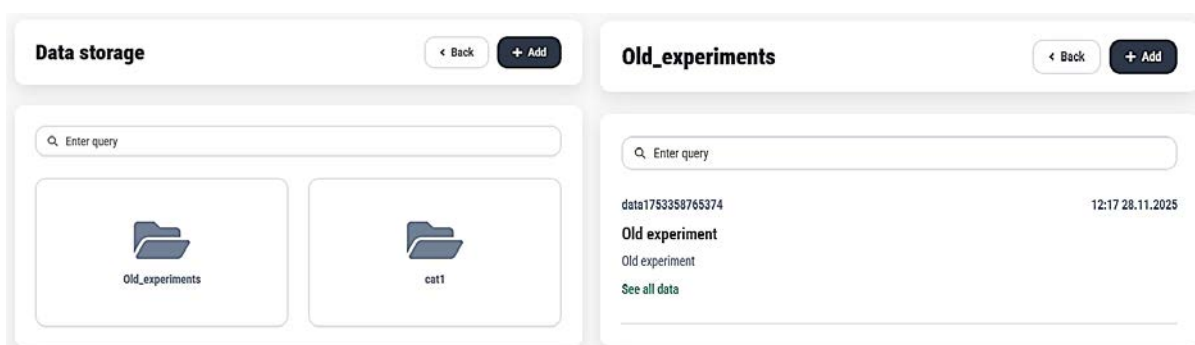


Рис. 8. Сторінки сховища даних експериментів

- updated – дата оновлення запису;
- email – електронна адреса;
- password – пароль;
- role – роль користувача;
- status – статус облікового запису;
- avatar – зображення профілю;
- bio – інформація про користувача;
- projects – проекти користувача;
- projectHistory – результати попередніх досліджень;

- activity – історія активності;
- friends – список пов’язаних користувачів.

Основні аутентифікаційні поля – id, email, password. Для управління доступністю та організації прав доступу застосовується поля status і role. Часові параметри змін акаунту зберігаються в created та updated. Окрім базових атрибутів, структура містить JSON-поля activity та bio, які містять структуру з відповідною інформацією. Поля projects та friends – це колекції з відповідними ідентифікаторами пов’язаних проектів та користувачів. Поле projectsHistory містить структуру з результатами попередніх досліджень, та є JSON-об’єктом, у якому ключами виступають ідентифікатори проектів, а значенням кожного ключа є масив записів виконань цього проекту. Кожний запис про виконання містить:

- id – унікальний ідентифікатор виконання;
- params – параметри експерименту, використаних під час запуску;
- results – результати виконання.

Колекція projects є центральною частиною структури. Вона об’єднує програмну реалізацію, налаштування інтерфейсу та механізми спільної роботи. Кожен проект містить:

- id – унікальний ідентифікатор проекту;

- author – ідентифікатор автора проекту (користувача);
- codeLang – мова програмування;
- date – дата останнього редагування користувачем;
- description – опис проекту;
- interface – структура елементів інтерфейсу;
- params – параметри запуску;
- shared – список користувачів із доступом до проекту;
- sourceCode – програмний код проекту;
- status – рівень відкритості проекту (приватний або публічний);
- title – назва проекту;
- created – дата створення;
- updated – дата останнього оновлення.

Особливістю структури є поле interface. Воно містить масив об’єктів, які описують елементи графічного інтерфейсу. Кожен елемент має поля name, description, type, value, valuesTemplate. Поле type визначає тип елемента керування (наприклад, select, single, multiple). Поле value зберігає поточне значення параметра, а valuesTemplate – перелік допустимих значень, якщо це потрібно. Таким чином, поле interface описує, як повинен виглядати інтерфейс параметризації досліджень. На його основі формується інтерфейс, а введені в нього параметри зберігаються у полі params, та використовуються програмним кодом з поля sourceCode. Це дозволяє виконувати дослідження з різними параметрами без зміни коду.

Колекція data призначена для збереження результатів експериментів, які використовуються як вхідні дані при проведенні досліджень (виконанні проектів). Вона містить поля:

- id – унікальний ідентифікатор запису;

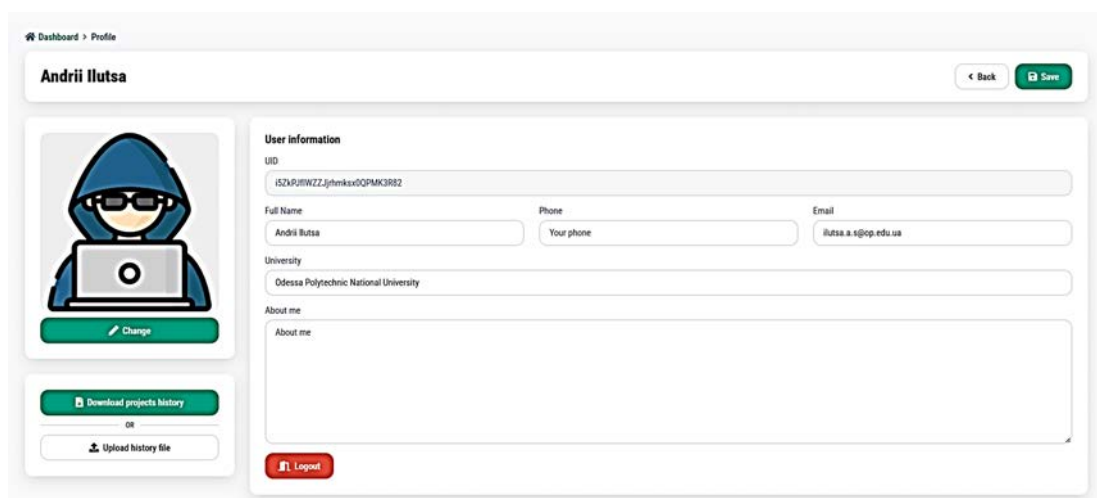


Рис. 9. Сторінка керування профілем користувача

- authors – список авторів експерименту;
- category – категорія даних;
- date – дата проведення експерименту;
- description – опис експерименту;
- isPublic – ознака відкритого доступу;
- title – назва;
- value – результати експерименту;
- created – дата створення запису;
- updated – дата останнього оновлення.

Поле value є універсальним контейнером для зберігання експериментальних даних. У ньому можуть зберігатися числові дані, масиви, вкладені об'єкти або інші параметри. Саме ці дані, у разі необхідності, використовуються як вхідні при виконанні досліджень. Таким чином, data виконує роль сховища експериментальних даних, на основі яких проводяться подальші дослідження.

Колекція logs забезпечує фіксацію запуску та виконання задач. Основні поля:

- id – унікальний ідентифікатор запису;
- err – інформація про помилку (якщо була);
- projectId – ідентифікатор проекту;
- taskId – ідентифікатор задачі виконання;
- uid – внутрішній ідентифікатор користувача;
- created – дата створення запису;
- updated – дата останнього оновлення.

Варто зазначити, що сам запис логування створюється під час запуску виконання проекту, а вже після отримання результатів він оновлюється, додаючи до себе ідентифікатор задачі виконання на сервері (taskId) та помилку (err), якщо така була.

У результаті, ця колекція дозволяє відстежувати процес виконання та аналізувати технічні збої.

Загалом, структура даних, наведена на рисунку 10, підтримує послідовний цикл дослідження:

1. Експериментальні дані зберігаються у колекції data.
2. Користувач user створює або налаштовує проект у projects, задаючи параметри через інтерфейс, створений на основі поля interface, які зберігаються у полі params.
3. Проект виконується з використанням даних із data.
4. Процес виконання фіксується у logs.
5. Результат дослідження зберігається у полі projectsHistory користувача user.

Таким чином, структура даних охоплює всі етапи роботи: накопичення вхідних експериментальних даних, їх обробку, отримання та збереження отриманих результатів, що забезпечує повний цикл дослідницької діяльності.

Висновки. Розроблено веб-орієнтовані програмні засоби, що поєднують сервіси PaaS і SaaS та забезпечують комплексну організацію хмарних обчислень у межах єдиного середовища, що підвищує продуктивність досліджень на освітнього процесу. Було обґрунтовано необхідність інтеграції, в рамках одного середовища, інструментів програмної розробки та готових інструментальних засобів (у вигляді GUI-інтерфейсів), оскільки їх розмежування в існуючих рішеннях знижує гнучкість, масштабованість і загальну ефектив-

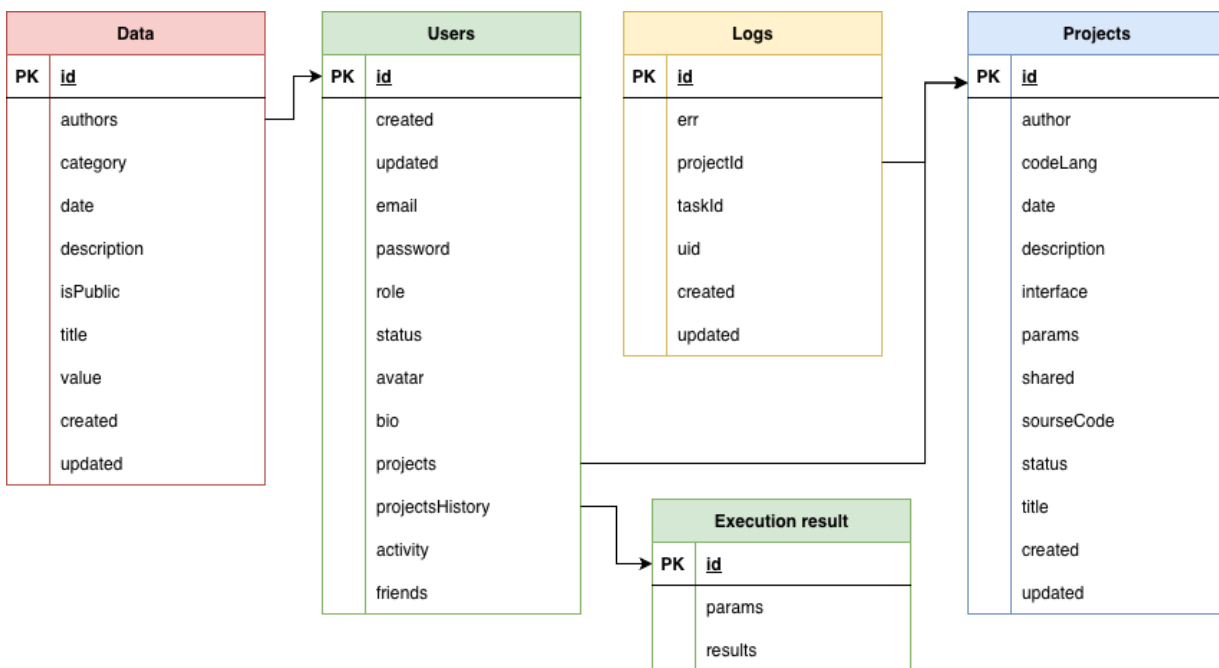


Рис. 10. Структура даних розроблених програмних засобів

ність міждисциплінарних досліджень та освітньої діяльності.

Проведений аналіз сучасних засобів організації хмарних обчислень показав, що вони не забезпечують комплексного поєднання роботи з даними та програмним кодом, створення GUI-інтерфейсів і підтримки спільної взаємодії користувачів у межах одного веб-орієнтованого рішення. Це обмежує можливості створення прикладних інструментів, ускладнює їх масштабування та повторне використання.

Розроблені програмні засоби реалізують інтегроване середовище, у якому забезпечено: збереження результатів експериментів та їх подальше використання, одночасну роботу з програмним кодом і графічними інтерфейсами, виконання обчислень на сервері, збереження та візуалізацію результатів досліджень. Ключовим результатом

є реалізація механізму взаємодії між інтерфейсом і програмною логікою завдяки конструктору інтерфейсів та транслятору коду, що дозволяє змінювати умови виконання досліджень без внесення змін до скрипт-коду. Такий підхід забезпечує відокремлення логіки реалізації від засобів керування параметрами виконання та підвищує зручність використання інструментів у міждисциплінарних дослідженнях. Це дозволяє розробляти інструментальні засоби та одночасно забезпечувати їх зручне використання без створення окремих застосунків для налаштування параметрів і без втручання у вихідний код під час зміни умов виконання. У результаті скорочується час підготовки та модифікації досліджень, спрощується адаптація інструментальних засобів до нових задач і підвищується продуктивність дослідницького та освітнього процесів.

Список літератури:

1. Opwonya J., Doan D.N.T., Kim S.G., Kim J., Ku B., Kim S., Park S., Kim J.U., Saccadic Eye Movement in Mild Cognitive Impairment and Alzheimer's Disease: A Systematic Review and Meta-Analysis, *Neuropsychology Review*, Vol.32, 2022, pp. 193–227. DOI: <https://doi.org/10.1007/s11065-021-09495-3>.
2. Jansson D., Rosén O., and Medvedev A., Non-parametric analysis of eye-tracking data by anomaly detection, *IEEE Transaction control system technology*, Vol.23, 2015, pp. 1578–1586. DOI: <https://doi.org/10.23919/ECC.2013.6669561>.
3. Bro V., Medvedev A., Continuous and Discrete Volterra-Laguerre Models with Delay for Modeling of Smooth Pursuit Eye Movements, *IEEE Transactions on Biomedical Engineering*, Vol.70, No.1, 2023, pp. 97–104. DOI: <https://doi.org/10.1109/TBME.2022.3185669>.
4. Lanata L., Sebastian L., Di Gruttola F., Di Modica S., Scilingo E.P., and Greco1 A., Nonlinear Analysis of Eye-Tracking Information for Motor Imagery Assessments, *Frontiers in Neuroscience*, Vol.13, 2020, pp. 1431. DOI: <https://doi.org/10.3389/fnins.2019.01431>.
5. Pavlenko V., Milosz M., Dzienkowski M., Identification of the oculo-motor system based on the Volterra model using eye tracking technology, 4th Int. Conf. on Applied Physics, *Simulation and Computing (APSAC'2020) 2020, 23-25 May, Rome, Italy. Journal of Physics: Conference Series 1603*, pp. 1–8. DOI: <https://doi.org/10.1088/1742-6596/1603/1/012011>.
6. Keehn B., Monahan P., Enneking B., Ryan T., Swigonski N., Keehn R.M., Eye-Tracking Biomarkers and Autism Diagnosis in Primary Care, *JAMA Netw Open*, Vol.7, No.5, 2024, pp. 1-14. DOI: <https://doi.org/10.1001/jamanetworkopen.2024.11190>.
7. Weiss K., Kolbe M., Lohmeyer Q., and Meboldt M., Measuring teamwork for training in healthcare using eye tracking and pose estimation, *Front. Psychol. Sec. Organizational Psychology*, Vol.14, 2023, pp. 1–12. DOI: <https://doi.org/10.3389/fpsyg.2023.1169940>.
8. Saraswat M., Tripathi R. C., Cloud computing: Analysis of top 5 CSPs in SaaS, PaaS and IaaS platforms, *9th International Conference System Modeling and Advancement in Research Trends (SMART)*, 2020, pp. 300-305. DOI: <https://doi.org/10.1109/SMART50582.2020.9337157>.
9. Kollipara P., An Overview on Cloud Computing, *International Journal of Research in Engineering, Science and Management*, Vol.4, No.8, 2021, pp. 35-37. URL: <https://journal.ijresm.com/index.php/ijresm/article/view/1147>.
10. King T. S., Reviews of Cloud Computing For Education: Services And Benefits, *PEOPLE: International Journal of Social Sciences*, Vol.1, No.1, 2015, pp. 1299-1305. DOI: <https://doi.org/10.20319/pijss.2015.s21.12991305>.
11. Singh A., Sharma S., Kumar S. R., Yadav S. A., Overview of PaaS and SaaS and its application in cloud computing, *International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*, 2016, pp. 172-176. DOI: <https://doi.org/10.1109/ICICCS.2016.7542322>.
12. Mohammed C. M., Zeebaree, S. R., Sufficient comparison among cloud computing services: IaaS, PaaS, and SaaS, *International Journal of Science and Business*, Vol.5, No.2, 2021, pp. 17-30. URL: <https://ijsab.com/wp-content/uploads/667.pdf>.

13. Wulf F., Lindner T., Strahringer S., Westner M., IaaS, PaaS, or SaaS? The why of cloud computing delivery model selection. Vignettes on the post-adoption of cloud computing, *Proceedings of the 54th Hawaii International Conference on System Sciences (HICSS-54)*, 4-9 January, Lahaina, Hawaii, USA, 2021, pp. 6285-6294. URL: https://aisel.aisnet.org/hicss-54/os/practice-based_research/5/.
14. Nadeem F., Evaluating and ranking cloud IaaS, PaaS and SaaS models based on functional and non-functional key performance indicators, *IEEE Access*, Vol.10, 2022, pp. 63245-63257. DOI: <https://doi.org/10.1109/ACCESS.2022.3182688>.
15. Project Jupyter. URL: <https://jupyter.org>.
16. Google Colab. URL: <https://colab.research.google.com>.
17. OnlineGDB. URL: <https://www.onlinegdb.com>.
18. MATLAB Online. URL: <https://www.mathworks.com/products/matlab-online.html>.
19. IONDS – Cloud computing platform. URL: <https://about.ionds.app>.
20. Павленко В.Д., Ілуща А.С., Гідулян В.І., Поєднання сервісів PaaS та SaaS на платформі для хмарних обчислень для застосувань у нейронауках, *Вісник ДНУ імені Олеся Гончара, Серія: Ракетно-космічна техніка*, 34(2), 2025, С. 131-139. DOI: <https://doi.org/10.15421/452522>.
21. Pavlenko V., Ilutsa A., Integration of PaaS and SaaS services in cloud computing software. *Monograph: "Intellectual and technological potential of the XXI century 2025"*, Vol.1, 2025, pp. 50–61. DOI: <https://doi.org/10.30890/2709-2313.2025-43-01>.
22. Pavlenko V., Lukashuk D., Ilutsa A., *Assessment of Psychophysiological Status Using a Cloud Computing Platform*, *SWorldJournal*, Vol.33, No.1, 2025, pp. 196–207. DOI: <https://doi.org/10.30888/2663-5712.2025-33-01-124>.

Ilutsa A.S. WEB-ORIENTED SOFTWARE INTEGRATION OF PAAS AND SAAS TO INCREASE THE PRODUCTIVITY OF RESEARCH AND EDUCATIONAL ACTIVITIES

The purpose of the work is to develop web-oriented software tools that combine PaaS and SaaS services and ensure increased efficiency of educational activities and interdisciplinary research. The problem of organizing cloud computing in the conditions of digital transformation of education and science is considered. It is shown that the use of local computing resources and the separation between software development environments and ready-made tools in the form of GUI interfaces reduces the flexibility, scalability and efficiency of research and educational processes. The analysis of existing solutions showed that they do not provide a comprehensive combination of opportunities for developing software code, creating interfaces and organizing joint work within a single environment. Web-oriented software tools have been developed that provide: simultaneous work with software code in the Python and JavaScript programming languages and GUI interfaces; performing calculations on the server; storing results and their visualization regardless of the programming language; the ability to integrate third-party sensors for conducting experiments and accumulating and reusing data from them; organization of joint user interaction. A mechanism for interaction between the interface and program logic has been implemented, due to the developed interface designer and code translator, which allows changing the conditions of research without editing the source code. Additionally, increased reproducibility of research results and adaptability to different subject areas are ensured due to modularity and flexible configuration of experimental processes, which is achieved by saving input data, execution parameters and research results. The proposed solution provides increased productivity of research and educational activities by integrating the development processes, performing calculations and using application tools in the form of GUI interfaces within a single web-oriented environment.

Keywords: web-platform, software, remote research, remote education, digital education, PaaS, SaaS, cloud technologies, cloud computing.

Дата першого надходження статті до видання: 17.03.2026

Дата прийняття статті до друку після рецензування: 15.04.2026

Дата публікації (оприлюднення) статті 11.05.2026