УДК 004.942
DOI: 10.32626/2308-5916.2024-25.46-62

В. Д. Павленко, д-р техн. наук, професор,
А. С. Ілуца, аспірант,
В. І. Гідулян, здобувач
Національний університет «Одеська політехніка», м. Одеса

WEB-ПЛАТФОРМА ДЛЯ ОРГАНІЗАЦІЇ ХМАРНИХ ОБЧИСЛЕНЬ У НЕЙРОФІЗІОЛОГІЧНИХ ДОСЛІДЖЕННЯХ НА ОСНОВІ ДАНИХ АЙТРЕКІНГУ

Метою роботи є розробка архітектури та web-версії програмного комплексу на основі запропонованої нової концепції організації хмарних обчислень, що дозволяє розширити діагностичні можливості інструментальних засобів модельно-орієнтованої інформаційної технології оцінювання нейрофізіологічного стану людини із використанням методів нелінійної динамічної ідентифікації окорухової системи за даними айтрекінгу. Пропонується концепція хмарних обчислень, що ґрунтується на поєднанні сервісів PaaS та SaaS в складі розробленого програмного комплексу, за рахунок чого забезпечується кросплатформеність хмарних обчислень, підвищується продуктивність і ефективність наукових досліджень. Розроблена архітектура дозволяє легко розширювати функціонал програмного комплексу та адаптувати його під різні умови застосування. Ключовою особливістю комплексу є його невимогливість до апаратного забезпечення на стороні клієнта завдяки використанню хмарних обчислень та його модульна структура, що дозволяє легко масштабувати комплекс, а також ізоляція процесу виконання скрипт-коду у середовищі хмарних обчислень для підвищення рівня безпеки при інтерпритації скрипт-коду на сервері. Порівняно з іншими подібними сервісами, комплекс має кілька переваг: він забезпечує ефективну роботу в дослідницьких і навчальних застосуваннях, підтримує мови програмування Python і JavaScript, а також дозволяє використовувати програмно-реалізовані методи ідентифікації через спеціально розроблені GUI-інтерфейси. Крім того, комплекс пропонує соціальні можливості та високий рівень абстракції, що дозволяє оптимізувати дослідницький процес.

Ключові слова: *web-платформа, хмарні сервіси, хмарні обчислення, PaaS, SaaS, модельно-орієнтована інформаційна технологія, ідентифікація, технологія айтрекінгу, нейрофізіологічні дослідження.*

Вступ. Вивчення взаємозв'язку окорухових функцій із центральною нервовою системою та аналіз психоемоційного стану людини сприяє кращому розумінню мозкових механізмів, їх порушень, дина-

міки психофізіологічних станів, а також процесів сприйняття, мислення, уяви та диференціації особистих намірів і установок. Технологія стеження за рухами очей (eye-tracking) сьогодні активно використовується в діагностичних дослідженнях нейрофізіологічних станів [1-4], вивчення когнітивних процесів і пам'яті [5], а також для моніторингу поведінки та навчання студентів [6]. Такі дослідження дають змогу глибше розуміти як свідомі, так і підсвідомі аспекти людської поведінки. Знання про рухи очей мають як теоретичне, так і прикладне значення, і вони відкривають нові можливості для дослідження особливостей різних професій з метою підвищення ефективності трудової діяльності. Широке використання апаратного забезпечення для інноваційної технології айтрекінгу в експериментальних дослідженнях нейронних процесів потребує розробки спеціалізованого програмного забезпечення для управління великими масивами даних [7-9]. Існує запит на надійні та точні показники психічного здоров'я як окремих осіб, так і груп населення, а також на обґрунтовані індикатори для моніторингу достовірності та валідності даних. Використання технологій розпізнавання емоцій дозволяє зробити висновки про стан нервової системи та оцінити її стан у повсякденних ситуаціях з підвищеним ризиком. Аналізуючи зміни траєкторій руху очей, можна робити конкретні висновки щодо психофізіологічного стану досліджуваних осіб. Для впровадження цієї технології в наукові дослідження різних інституцій та навчальних закладів доцільно реалізувати її з використанням хмарних сервісів.

Хмарні сервіси стали однією з основних рушійних сил сучасної цифрової трансформації. Вони забезпечують доступ до обчислювальних ресурсів, таких як сервери, сховища даних, бази даних, мережі, програмне забезпечення, аналітичні інструменти та багато іншого через Інтернет [10-12]. Ці ресурси можуть бути надані в різних моделях, що дозволяє користувачам оптимізувати використання IT-ресурсів, зменшуючи витрати та підвищуючи ефективність.

Хмарні обчислення пропонують кілька різних моделей надання послуг, кожна з яких має свої переваги та підходить для різних потреб:

- **Software as a Service (SaaS).** Це найпоширеніша модель, яка дозволяє користувачам отримувати доступ до програмного забезпечення через Інтернет. Користувачі можуть працювати з додатками без необхідності їх встановлення або обслуговування на локальних пристроях.
- **Platform as a Service (PaaS).** PaaS забезпечує середовище для розробки, тестування та розгортання додатків, зменшуючи складність управління інфраструктурою.
- **Infrastructure as a Service (IaaS).** IaaS надає віртуальні обчислювальні ресурси, такі як сервери, сховища даних, мережі та інші базові

обчислювальні елементи, які користувачі можуть використовувати для створення та управління своїми власними IT-системами.

- **Function as a Service (FaaS):** FaaS є підвидом PaaS, де користувачі можуть запускати функції або частини коду у відповідь на певні події, не турбуючись про управління серверами.

Хмарні технології призначені для консолідації IT-інфраструктур, аутсорсингу IT-ресурсів, а також об'єднання обчислювальних ресурсів на базі серверів, сховищ, мереж і програм зі спільним доступом та забезпечують динамічне масштабування, гнучкість, низькі витрати та доступність, проте мають наступні виклики сфері безпеки [13-14]:

- **Забезпечення безпеки даних.** Оскільки дані зберігаються та обробляються на віддалених серверах, підвищується ризик витоку даних, несанкціонованого доступу та інших кіберзагроз. Користувачі повинні впевнитися, що їхні дані надійно захищені, а провайдери дотримуються найкращих практик кібербезпеки.
- **Залежність від Інтернет-з'єднання.** Хмарні обчислення вимагають надійного та стабільного Інтернет-з'єднання. У разі перебоїв у роботі Інтернету користувачі можуть втратити доступ до критичних ресурсів і даних.
- **Контроль та управління.** Використання хмарних обчислень може обмежувати контроль користувачів над інфраструктурою та операціями, які виконуються в хмарі. Це може стати проблемою, якщо виникають проблеми, що потребують швидкого втручання, або якщо потрібно забезпечити спеціальні налаштування для певних додатків.
- **Правові та регуляторні питання.** Передача даних у хмару може створювати правові виклики, особливо якщо дані переміщуються між різними країнами з різними законами про захист даних. Організації повинні ретельно вивчати правові аспекти використання хмарних послуг, щоб уникнути порушень регуляторних вимог.

Хмарні обчислення є однією з найважливіших технологій сучасності, яка швидко розвивається і трансформує різні галузі [15-18]. Вони забезпечують масштабованість, гнучкість у різних наукових напрямках та економічну вигоду для бізнесу, дозволяючи скорочувати витрати на IT-інфраструктуру, підвищувати продуктивність та ефективно управляти даними. Попри наявні виклики, такі як питання безпеки та приватності, хмарні обчислення продовжують розвиватися, інтегруючись з іншими передовими технологіями та пропонуючи нові можливості для інновацій. З огляду на це, важливо продовжувати дослідження у цій сфері, щоб максимально використовувати потенціал хмарних технологій у майбутньому.

Таким чином, хмарні обчислення стали незамінним інструментом у сучасному бізнесі та науці, відкриваючи нові горизонти для розвитку і вдосконалення процесів у різних сферах діяльності.

У даний час для підтримки хмарних обчислень застосовуються такі сервіси як Jupyter [19] і Google Colab [20]. Це більше інтерактивні блокноти, ніж платформи для проведення експериментів і роботи з результатами обробки даних. Вони використовуються як редактори та середовища виконання для однієї мови програмування, не забезпечують можливість працювати над програмним кодом проєктів на декількох мовах програмування одночасно та не надають можливості взаємодіяти з вже реалізованими інтерфейсами для хмарних обчислень у нейрофізіологічних дослідженнях на основі даних айтрекінгу.

Існує потреба у створені нової концепції організації хмарних обчислень у нейрофізіологічних дослідженнях на основі даних айтрекінгу та програмного комплексу/платформи що підтримуватиме хмарні обчислення одразу двох видів PaaS і SaaS. Це дозволить ефективно працювати у дослідницьких і навчальних напрямках як з програмним кодом на декількох мовах програмування Python, JavaScript для удосконалення алгоритмів тощо, так і з реалізованими раніше методами ідентифікації у вигляді GUI інтерфейсів. Також, зважаючи на великий об’єм даних для проєктів на даній платформі та їх можливу залежність один від одного, важливою частиною є соціальна складова, яка дозволить спростити обмін даними між науковцями та між проєктами, підвищити продуктивність наукових досліджень, чого немає у схожих існуючих рішеннях.

Метою роботи є розробка архітектури та web-версії програмного комплексу на основі запропонованої нової концепції організації хмарних обчислень, що дозволяє розширити діагностичні можливості інструментальних засобів модельно-орієнтованої інформаційної технології оцінювання нейрофізіологічного стану людини із використанням методів нелінійної динамічної ідентифікації окорухової системи за даними айтрекінгу.

Концепція організації хмарних обчислень. Запропоновано нову концепцію організації хмарних обчислень яка полягає у об’єднанні двох видів взаємодії платформи з користувачем: PaaS та SaaS (рис. 1). Нова концепція дає можливість організувати хмарні обчислення таким чином, що завдяки вбудованому інтерфейсу та редактору програмного коду є можливість здійснювати дослідження одночасно як на рівні програмного коду так і на рівні інтерфейсу, задаючи параметри досліджень.

Нова концепція дозволить проводити дослідницьку та освітню діяльність, модифікувати проєкти на рівні програмного коду в рамках одного програмного комплексу. Це дозволяє спростити та оптимізувати процес обміну інформацією та результатами досліджень.

Архітектура програмного комплексу. Розроблено концепцію, архітектуру та веб-орієнтований комплекс/платформу, що забезпечує інтерфейс та функціональність для хмарних обчислень за наступними

принципами роботи: платформа як сервіс PaaS та програмне забезпечення як сервіс SaaS.

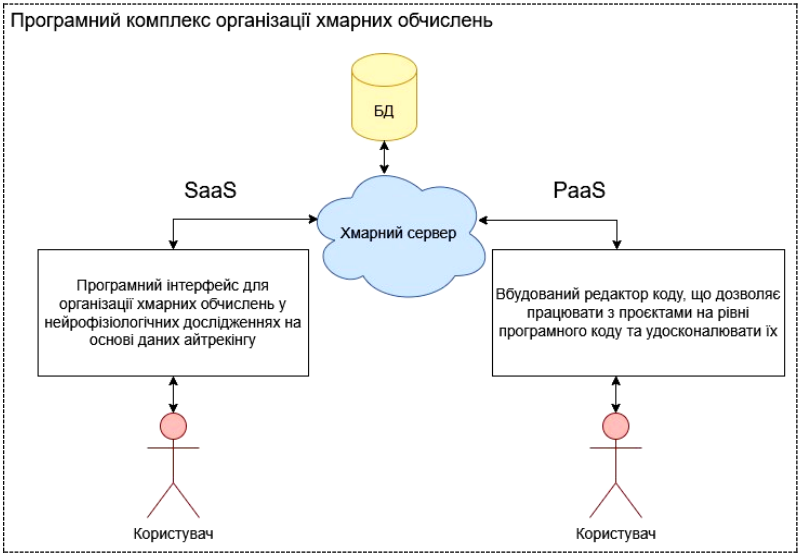

Рис. 1. *Концепція організації хмарних обчислень*

Комплекс дозволяє автоматизувати дослідження для ідентифікації нелінійних динамічних систем і додавання нових методів за допомогою вбудованого редактора коду (PaaS-сервіс). Крім того, він надає можливість редагувати та виконувати код скрипта будь-якого методу ідентифікації (якщо код організований відповідно до документації), додавати до коду список параметрів експерименту, виконувати обчислення за допомогою інтегрованого скрипт-коду на сервері, збирати результати у браузері на стороні клієнта і зберігати їх. Комплекс дозволяє здійснювати спеціальну підготовку даних експериментальних досліджень око-рухової системи (ОРС), отриманих від айтрекера, для подальшої обробки в процедурі нелінійної динамічної ідентифікації (SaaS-сервіс). Процес підготовки даних складається з наступних етапів:

Етап 1: Очищення даних:

- зчитування даних, наданих айтрекером;
- визначення та виділення з потоку даних окремих циклів досліджень для кожного респондента;
- видалення артефактів;
- витяг числових масивів відгуків на тестові візуальні стимули;
- відокремлення фрагментів – масивів відповідей на окремі стимули;
- видалення фіксацій різної тривалості на перехідному процесі.

Етап 2: Передобробка даних:

- перетворення послідовності фрагментів у паралельну структуру та приведення відгуків в циклі досліджень до спільного початку;
- нормалізація даних.

Етап 3: Процедура ідентифікації:

- проведення процедури ідентифікації ОМС за даними, отриманими після передобробки.

Розроблені програмні засоби являють собою програмний комплекс, що складається з багатьох модулів і вузлів, які взаємодіють між собою. Нижче наведено схематичне зображення структури програмного комплексу (рис. 2).

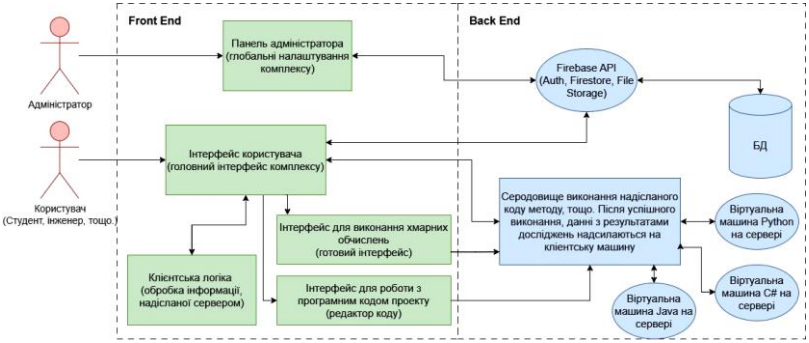

Рис. 2. *Структура програмного комплексу*

Панель адміністратора програми – інтерфейс користувача для базової конфігурації платформи:

- управління обліковими записами користувачів;
- управління загальною службовою інформацією.

Ця частина підключається до Firebase API для збереження змін і відображення інформації.

Інтерфейс користувача (основний інтерфейс програми) – інтерфейс користувача для взаємодії з платформою та використання її функцій:

- додавання нових методів ідентифікації;
- редагування існуючих методів відповідно до прав доступу та редагування;
- зміна даних облікового запису;
- додавання співавторів і обмін методами ідентифікації з ними; збереження та обмін результатами експерименту.

Також він містить вбудований редактор коду (інтерфейс для завантаження нових методів ідентифікації), що дозволяє ефективно

працювати з кодом методів ідентифікації. Підключається до Firebase API для збереження змін і відображення інформації.

Firebase API – ця частина служби обробляє всі клієнтські запити, і надсилає запити до бази даних або служби, яка використовується як база даних (бібліотека Firebase). Незалежність цієї частини комплексу дозволяє з легкістю додавати клієнти з інших платформ (Windows, MacOS, Linux, Android, iOS та ін.).

Середовище виконання коду – ізольована частина програмного комплексу, яка діє як програма для виконання коду для Python, JavaScript, на якій виконуються сценарії всіх методів ідентифікації. Потім результати надсилаються на клієнтську частину для подальшого ознайомлення і, якщо потрібно, зберігання в базі даних. Для інтерпретації коду використовується віртуальна машина Python, JavaScript.

Бібліотека Firebase – бібліотека компонентів, яка через свій API дозволяє працювати з noSQL базою даних Firebase Firestore для зберігання даних користувача. Він також підтримує Firebase Auth для аутентифікації користувачів і контролю доступу, а також Firebase Storage для зберігання файлів.

Firebase Auth – модуль для аутентифікації користувачі у комплексі. Він підтримує вендорів соціальних мереж, таких як Facebook, GitHub, Twitter, Google і Google Play Games, і містить систему керування користувачами для аутентифікації користувачів за допомогою електронної пошти та пароля, які зберігаються у Firebase.

Firebase Storage забезпечує надійне завантаження та вивантаження файлів для застосунків Firebase незалежно від якості мережі. Розробник може використовувати його для зберігання зображень, аудіо-, відео- чи іншого вмісту, створеного користувачами. Зберігання Firebase підтримується Google Cloud Storage.

Розроблений комплекс – це модульний програмний комплекс, що складається з:

- незалежна серверна частина, яка включає всі обчислювальні модулі, базову логіку та обробку даних. Кожен модуль окремий і може масштабуватися незалежно;
- клієнтська частина, яка представляє інтерфейс користувача та клієнтську логіку і є кросплатформеним веб-інтерфейсом, що представляє з себе SPA (односторінковий додаток).

У той же час, структурно та з точки зору взаємодії функціональних компонентів програмний продукт складається з 4-х основних компонентів функціональної схеми роботи сервісу (рис. 3):

- сервер, що займається управлінням усіма технічними процесами: приймання запитів, надання відповідей на запити, формування завдань агентам, збереження інформації у базі даних;

- веб-інтерфейс, що надає користувачу можливість користуватися сервісом та працювати з проєктами;
- агенти – обчислювальні компоненти які виконують поставлені завдання, інтерпретують надісланий код;
- база даних для збереження усіх даних комплексу та проєктів.

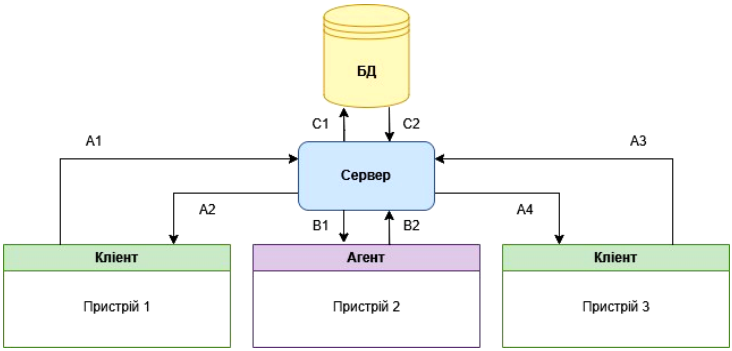

Рис. 3. *Функціональна схема роботи сервісу*

На схемі зображено усі можливі варіанти спілкування структурних елементів сервісу між собою, а саме:

A1, A3 – Клієнт відправляє завдання серверу;
B1 – Сервер відправляє завдання агенту;
B2 – Агент відправляє результат серверу;
A2, A4 – Сервер відправляє результат/данні клієнту;
C1 – Сервер зберігає інформацію у базі даних;
A4 – Сервер отримає інформацію з бази даних.

Одним з основних є процес виконання скрипт-коду проєкту. Цей процес і процес надання результатів, що відбувається на клієнті, можна описати наступним псевдокодом:

```
algorithm execute_code_on_server is
input: project
if (is_running) return
if (!project.code) return
out := null
err := null
is_running := true
out, err := request('POST /',
project.code, project.files)
is_running := false
if (err)
show_errors(err)
return
```

```
graphs = parse(out)
show_results(graphs)
```

Процес, що відбувається на сервері для надання результатів, може бути описаний таким псевдокодом:

```
algorithm server is
on POST / do
uid := generate_unique_id()
project_path := get_project_path(uid)
create_project_dir(project_path)
main_path := project_path + "main.ext"
save_file(request.code, main_path)
for each file in request.files do
save_file(file, project_path)
out, err := execute(main_path)
return uid, out, err
```

Така архітектура дозволяє незалежно вести розробку та масштабування обох частин комплексу. Що дозволить у майбутньому додати до комплексу мобільні додатки та інші пропрієтарні додатки для будь яких платформ. Також, завдяки тому, що вся логіка та обробка даних знаходиться у серверній частині проєкту, створено API-інтерфейс, який дозволить стороннім додаткам використовувати функціонал розробленого програмного комплексу. Це надає можливість ще більше масштабувати проєкт.

Опис інтерфейсу та процеса роботи з комплексом. Процес роботи з програмним комплексом для отримання результатів складається з наступних кроків, які відображені на схемі (рис. 4).

Як показано на схемі, модулі працюють незалежно, що дозволяє масштабувати систему більш ефективно, без взаємного впливу модулів. При необхідності до серверних модулів можуть бути підключені альтернативні клієнти для різних платформ або завдань. Водночас, сервер може бути доповнений новими модулями без потреби вносити зміни в існуючих клієнтів.

На сторінці входу користувачі можуть отримати доступ до функцій комплексу, ввівши своє ім'я користувача та пароль. На початковому етапі роботи, функція реєстрації буде недоступна для користувачів, і всі облікові записи створюватиме адміністратор системи (супер-адмін). Це рішення прийнято для посиленого контролю над базою користувачів під час закритого тестування. Також, для доступу до функцій комплексу користувачам доступна авторизація за допомогою Google-провайдера. Це дозволить підвищити рівень зручності для користувачів за рахунок відсутності необхідності проходити процес реєстрації та покращити контроль над студентськими акаунтами за рахунок того, що корпоративні акаунти студентів, найчастіше, є Google-акунтами.

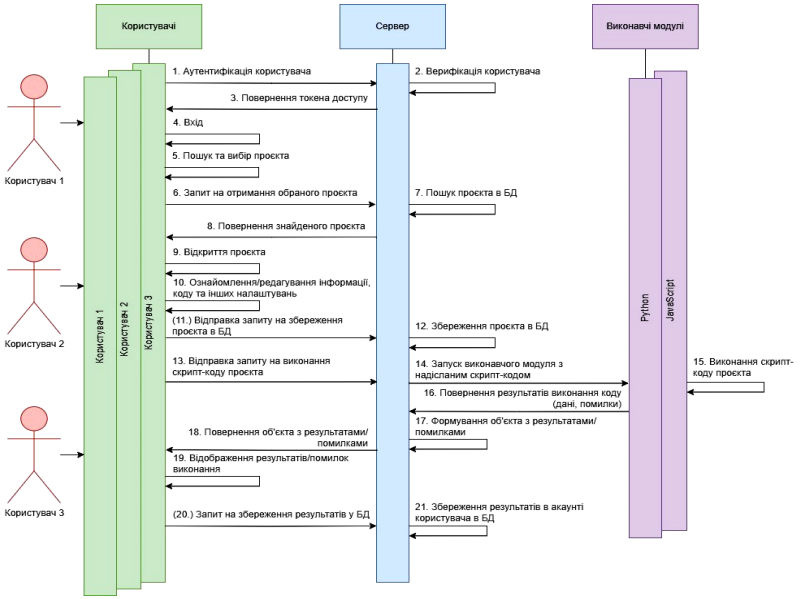

Рис. 4. *Процес роботи з програмним комплексом*

Головна сторінка комплексу виконує роль інформаційної панелі, яка консолідує всі ключові дані для користувача: останні події в проєкті, інформацію про облікові записи, обраних користувачів та останні проєкти. Крім того, на цій сторінці користувачі можуть перейти до повного списку обраних користувачів та проєктів, натиснувши відповідні посилання або кнопки. Для перегляду інформації про обраного користувача необхідно натиснути на його ім'я, після чого буде відображено ту ж інформаційну панель, але вже з даними обраного облікового запису (якщо доступ не обмежений).

Сторінка користувача призначена для перегляду та редагування особистої інформації, такої як: фото (аватар), ім'я, номер телефону, електронна пошта, навчальний заклад, тощо.

Сторінка проєктів надає доступ до публічних проєктів та особистих проєктів користувача. Карта проєкту, яка відображається на сторінці, містить короткий опис: назву, дату останнього редагування, автора та короткий огляд проєкту. На цій сторінці користувачі можуть створювати нові публічні або приватні проєкти. За замовчуванням проєкт створюється як приватний і недоступний для інших користувачів, але його можна зробити публічним у налаштуваннях. Розділ над списком проєктів дозволяє фільтрувати проєкти, відображаючи їх таким чином: всі загальнодоступні проєкти та приватні, доступні тільки для користувача. Будь-який користувач може переглядати та запускати скрипти публічних проєктів, але редагувати їхній скрипт-код або інформацію неможливо. Щоб працювати

з загальнодоступним проєктом як зі своїм власним, користувачу потрібно скопіювати його до свого профілю. Це дозволить повністю редагувати вихідний код проєкту, його дані та працювати з результатами.

Натиснувши на проєкт, користувач переходить на його сторінку. Сторінка проєкту надає можливість переглядати та керувати проєктом, зокрема:

- переглядати опис проєкту, його код та результати минулих досліджень (якщо вони є);
- редагувати код проєкту за наявності відповідних прав доступу (якщо користувач не є власником, то доступ може бути обмежений);
- змінювати рівень доступу проєкта (публічний або приватний);
- обирати мову скрипт-коду проєкта;
- запускати виконання скрипт-коду проєкту;
- аналізувати результати досліджень;
- надавати доступ до проєкту (коду та результатів дослідження) обраним користувачам.

Інтерфейс програмного комплексу розроблено с урахуванням усіх сучасних тенденцій та вимог UI/UX дизайну та адаптовано під усі види мобільних пристроїв: смартфони, планшетні комп'ютери тощо. Це можна побачити на прикладі сторінки проєкту на мобільному пристрої (рис. 5). Також, інтерфейс комплексу оснащений альтернативним режимом відображення, який представляє з себе темну тему оформлення, що разом с адаптованою версією інтерфейсу дозволить підвищити рівень зручності при роботі з ним, особливо для користувачів з певними вадами зору.

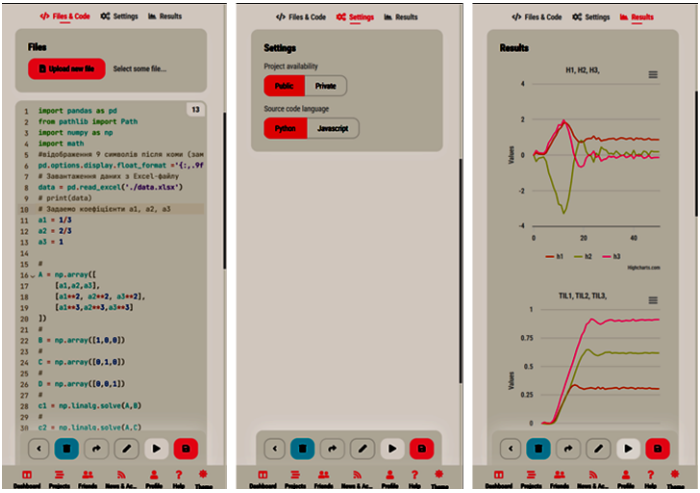

Рис. 5. *Складові сторінки проєкту на мобільному пристрої*

Ізоляція виконання скрипт-коду користувача у середовищі хмарних обчислень. В сучасних хмарних обчисленнях важливо безпечно та ефективно виконувати надісланий користувачем код. Це особливо важливо в середовищах, які підтримують кілька мов програмування, де користувачі очікують виконання коду в реальному часі через веб-інтерфейс. Враховуючи ненадійну природу надісланого користувачем коду, забезпечення надійної безпеки, ефективної ізоляції виконання скрипт-коду у середовищі хмарних обчислень та керування ресурсами має вирішальне значення (рис. 6). У цьому контексті контейнеризація процесів є хорошим рішенням для вирішення цих проблем, а також спрощує налаштування та керування різноманітними середовищами виконання.

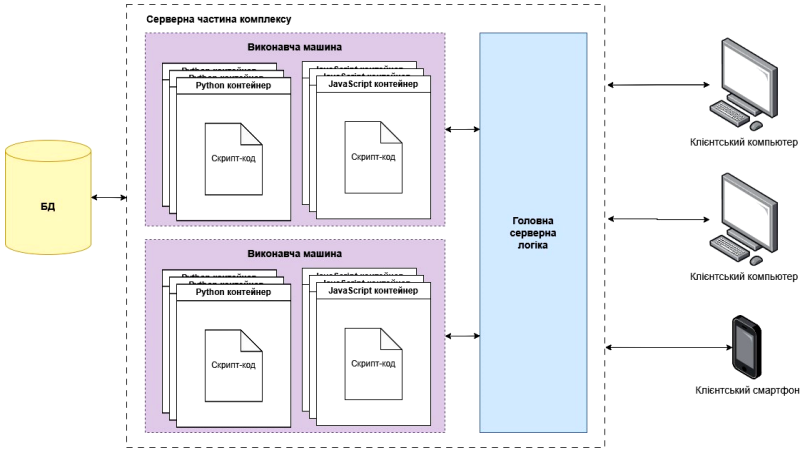

Рис. 6. *Принцип ізоляції виконання скрипт-коду у середовищі хмарних обчислень*

Для реалізації ізоляції процесів виконання скрипт-коду на сервері обрано контейнеризацію з використанням інструментарію для управління ізольованими Linux-контейнерами – Docker [21].

Контейнеризація процесу виконання скрипт-коду проєктів надає наступні переваги:

- **Безпека.** Виконання ненадійного коду користувача створює значні ризики для безпеки, особливо коли такий код працює на спільній інфраструктурі. Критичною проблемою є можливість доступу зловмисного коду до конфіденційних даних, що належать іншим користувачам або самій системі. Контейнери пропонують рівень безпеки, ізолюючи кожне середовище виконання. Кожен контейнер функціонує як середовище ізольованого програмного середовища з власною файловою системою, процесами та мережевим

стеком, що запобігає взаємодії коду користувача з хост-системою чи іншими контейнерами. Завдяки обмеженню можливостей контейнерів поверхня атаки зведена до мінімуму, гарантуючи, що навіть якщо користувач спробує використати систему, будь-яка шкода міститься в ізольованому середовищі.

- **Ізоляція.** Ізоляція є фундаментальною вимогою для виконання коду користувача в середовищі з кількома клієнтами. Технологія контейнеризації забезпечує ізоляцію процесів і ресурсів. Кожен контейнер працює у власному ізольованому середовищі, гарантуючи, що код, який виконується в ньому, не заважає або не отримує доступу до контейнерів інших користувачів або хост-системи. Ця ізоляція досягається за допомогою просторів імен і контрольних груп, основних функцій ядра Linux, які використовує Docker. Простори імен забезпечують ізольовані перегляди системних ресурсів, таких як дерева процесів, мережеві інтерфейси та файлові системи, тоді як контрольні групи керують і обмежують використання ресурсів для кожного контейнера. Ізольоване середовище в контейнері дозволяє одночасно виконувати кілька сценаріїв користувача без ризику того, що сценарій одного користувача вплине на сценарій іншого. Це має вирішальне значення для підтримки цілісності та надійності служби та забезпечення узгодженого досвіду для всіх користувачів. У випадку, якщо сценарій користувача поводиться непередбачувано, він буде міститися в призначеному контейнері, запобігаючи його впливу на ширшу систему або навантаження інших користувачів.
- **Управління ресурсами.** Управління споживанням ресурсів є ще одним важливим аспектом виконання коду користувача в хмарному середовищі. Контейнери дозволяють точно контролювати розподіл ресурсів, наприклад використання ЦП, обмеження пам'яті та дисковий ввід/вивід. Це особливо важливо при роботі з потенційно ресурсомісткими сценаріями, які можуть монополізувати ресурси сервера та знизити продуктивність для інших користувачів. Установлюючи обмеження на використання процесора та пам'яті, адміністратори можуть переконатися, що жоден контейнер не споживає більше, ніж належна йому частка ресурсів, таким чином зберігаючи стабільність і продуктивність системи.
- **Гнучкість у підтримці кількох мов програмування.** Однією з значних переваг використання Docker для виконання коду користувача є легкість налаштування різноманітних середовищ виконання та керування ними. Docker дозволяє створювати контейнери, адаптовані до певних мов програмування, кожен із яких має власний набір залежностей, бібліотек і конфігурацій середовища виконання. Цей підхід гарантує, що код користувача виконується

в узгодженому та відтворюваному середовищі. Крім того, проектуючи ці контейнери з уніфікованим інтерфейсом для обробки коду, впровадження нових мов програмування стає простим процесом. Після створення шаблону контейнера для певної мови додавання підтримки додаткових мов вимагає мінімальних зусиль.

Використання контейнеризації для виконання коду користувача в середовищі хмарних обчислень пропонує значні переваги з точки зору безпеки, ізоляції, управління ресурсами та гнучкості. Технологія контейнеризації не тільки забезпечує надійну безпеку та ізоляцію, але й спрощує налаштування та керування різноманітними середовищами виконання. Здатність застосовувати точні засоби керування ресурсами та легко розширювати підтримку нових мов програмування робить Docker ідеальним рішенням для безпечного та ефективного запуску ненадійного коду в хмарних веб-додатках. Ці функції спільно сприяють створенню більш стійкої та масштабованої системи, здатної задовольнити вимоги сучасних хмарних обчислень.

Наукова новизна. Вперше запропоновано дослідження з діагностування нейрофізіологічного стану людини на основі модельно-орієнтованої інтелектуальної інформаційної технології та експериментальних даних айтрекінгу, здійснюване за підтримки технології хмарних обчислень, що значно підвищує продуктивність та ефективність наукових досліджень, при участі в процесі діагностування багатьох дослідників з довільною локацією.

Отримала подальший розвиток технологія хмарних обчислень на основі запропонованої нової концепції організації хмарних сервісів працюючих одночасно за принципами: платформа як сервіс PaaS та програмне забезпечення як сервіс SaaS; що має переваги по відношенню до відомих подібних платформ, спрощує дослідницькі та освітні процеси та забезпечує невимогливість до апаратного забезпечення та кросплатформеність розробленого програмного комплексу.

Висновки. Розроблено архітектуру та програмний комплекс на основі запропонованої нової концепції організації хмарних обчислень, що розширює діагностичні можливості інструментальних засобів модельно-орієнтованої інформаційної технології оцінювання нейрофізіологічного стану людини, забезпечує кросплатформеність процесу наукових досліджень з використанням методів непараметричної ідентифікації досліджуваної око-рухової системи за даними айтрекінгу, підвищує їх продуктивність та ефективність.

Комплекс забезпечує можливість ефективно працювати у дослідницьких і навчальних напрямках як з програмним кодом на декількох мовах програмування Python, JavaScript для удосконалення алго-

ритмів тощо, так і з реалізованими раніше методами ідентифікації у вигляді GUI інтерфейсів та працювати з платформою на будь-якому пристрої як з хмарним сервісом. Можливість ефективно працювати з розробленим комплексом на різних пристроях досягається, у тому числі, завдяки спеціально адаптованому інтерфейсу, який змінюється в залежності від пристрою, на якому він відображається, та дозволяє зручно взаємодіяти з інформацією незалежно від розмірів пристрою.

Важливою особливістю розробленого програмного комплексу є невимогливість до апаратного забезпечення на клієнтській стороні, а також одночасне поєднання сервісів PaaS і SaaS та модульна структура програмного комплексу, що дозволяє легко здійснювати його масштабування. Програмний комплекс має переваги по відношенню до таких сервісів як Project Jupyter, Google Colub, а саме: можливість працювати з популярними мовами програмування та готовими методами у вигляді GUI інтерфейсів для досліджень, а також підвищення рівня абстракції та соціалізації, що оптимізує дослідницький процес та надає нові можливості для його користувачів. Комплекс має підтримку виконання скриптів методів на таких мовах програмування, як: Python, JavaScript. При розробці програмних засобів були використані: мова програмування JavaScript; мови HTML та CSS для створення інтерфейсу, фреймворк Vue.js [22] для мови JavaScript; Python для реалізації методів нелінійної динамічної ідентифікації; Node.js для реалізації клієнт-серверної взаємодії комплексу.

Список використаних джерел:

1. Opwonya J., Doan D. N. T., Kim S. G. et al. Saccadic Eye Movement in Mild Cognitive Impairment and Alzheimer's Disease: A Systematic Review and Meta-Analysis. *Neuropsychol Rev*. 2022. Vol. 32. P. 193-227. URL: https://doi.org/10.1007/s11065-021-09495-3.
2. Jansson D., Rosén O., Medvedev A. Parametric and nonparametric analysis of eyetracking data by anomaly detection. *IEEE Transaction control system technology*. 2015. Vol. 23. P. 1578-1586. DOI: 10.1109/TSCT.2014.2364958.
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.* 2023. Vol. 70 (1). P. 97-104.
4. Lanata L., Sebastian L., Di Gruttola F., et al. Nonlinear Analysis of Eye-Tracking Information for Motor Imagery Assessments. *Frontiers in Neuroscience.* 2020. Vol. 13:1431. DOI: 10.3389/fnins.2019.01431.
5. Keehn B., Monahan P., Enneking B. et al. Eye-Tracking Biomarkers and Autism Diagnosis in Primary Care. *JAMA Netw Open.* 2024. Vol. 7 (5): e2411190. P.1-14. DOI: 10.1001/jamanetworkopen.2024.11190.
6. Weiss K., Kolbe M., Lohmeyer Q., Meboldt M. Measuring teamwork for training in healthcare using eye tracking and pose estimation. *Front. Psychol.* 2023. Vol. 14: 1169940. P. 1-12. URL: https://doi.org/10.3389/fpsyg.2023.1169940.

7. Pavlenko V., Shamanina T., Chori V. Eyetracking Technology and its Application in Neuroscience. *Proceedings of the The 12th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS-2023)*, 7-9 Sept., Dortmund, Germany. 2023. Vol. 1. P. 187-193.
8. Pavlenko V., Shamanina T., Chori V. Biometric Identification based on the Multidimensional Transient Functions of the Human Oculo-Motor System. *5th International Conferenceon Applied Physics, Simulation and Computing (APSAC-2021), 3-5 Sept., Salerno, Italy. J. Phys.: Conf. Ser.* 2022. Vol. 2162, 012024. P. 1-9.
9. Pavlenko V. D., Shamanina T. V., Chori V. V. Estimation Psychophysiological State via Integral Nonlinear Model of the Oculo-Motor System. *Applied Aspects of Information Technology*. 2023. Vol. 6. № 2. P. 117-129. URL: https://doi.org/10.15276/aait.06.2023.8.
10. Яремчук Д. С. Сервіси і додатки хмарних обчислень. *Збірник тез IV Міжнародної науково-практичної інтернет-конференції «Цифрова трансформація фінансової системи України та країн v-4 в умовах євроінтеграції»*. 2024. С. 95-97.
11. 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. P. 300-305.
12. Kollipara P. An Overview on Cloud Computing. *International Journal of Research in Engineering, Science and Management*. 2021. Vol. 4 (8). P. 35-37.
13. Вихрист О. В., Петрова Р. В. Хмарні обчислення: переваги та недоліки. *Збірник матеріалів IX Міжнародної молодіжної науково-практичної інтернет-конференції.* 2023. С. 314-316.
14. Fatima E., Sumra I. A., Naveed R. A Comprehensive Survey on Security Threats and Challenges in Cloud Computing Models (SaaS, PaaS and IaaS). *Journal of Computing & Biomedical Informatics.* 2024. Vol. 7 (01). P. 537-544.
15. Yoo S. K., Kim B. Y. The effective factors of cloud computing adoption success in organization. *The Journal of Asian Finance, Economics and Business.* 2019. Vol 6 (1). P. 217-229.
16. Досенко А. К. Хмарні технології: прикладні технології сучасних платформ. *Вчені записки ТНУ імені ВІ Вернадського.* 2022. Вип. 33 (72). С. 257-262.
17. Грищук А., Хімко Я. Хмарні технології: поняття, особливості використання. *Збірник матеріалів VII Всеукраїнської науково-практичної конференції «Теоретико-прикладні проблеми правового регулювання в Україні»*. 2023. С. 137-139.
18. Смірнова Т. В., Поліщук Л.І., Смірнов О.А.та ін. Дослідження хмарних технологій як сервісів. *Кібербезпека: освіта, наука, техніка.* 2020. Т. 3. № 7. С. 43-62.
19. Project Jupyter. URL: https://jupyter.org.
20. Google Colub. URL: https://colab.research.google.com.
21. Docker. URL: https://www.docker.com.
22. Vue.js framework. URL: https://vuejs.org.

WEB-PLATFORM FOR CLOUD COMPUTING IN NEURO-PHYSIOLOGICAL RESEARCH BASED ON EYE-TRACKING DATA

The purpose of the work is to develop the architecture and web version of the software complex based on the proposed new concept of cloud computing organization, which allows to expand the diagnostic capabilities of the tools of model-oriented information technology for assessing the neurophysiological state of a person using methods of nonlinear dynamic identification of the oculomotor system based on eye tracking data. The concept of cloud computing is proposed, which is based on the combination of PaaS and SaaS services as part of the developed software complex, due to which the cross-platform nature of cloud computing is ensured, the productivity and efficiency of scientific research increases. The developed architecture allows you to easily expand the functionality of the software complex and adapt it to different application conditions. The key feature of the complex is its undemanding hardware on the client side thanks to the use of cloud computing and its modular structure, which allows for easy scaling of the complex, as well as the isolation of the script code execution process in the cloud computing environment to increase the level of security when interpreting the script code on the server . Compared to other similar services, the complex has several advantages: it provides effective work in research and educational applications, supports Python and JavaScript programming languages, and also allows the use of software-implemented identification methods through specially developed GUI interfaces. In addition, the complex offers social opportunities and a high level of abstraction, which allows to optimize the research process.

Key words: *web platform, cloud services, cloud computing, PaaS, SaaS, model-oriented information technology, identification, eye tracking technology, neurophysiological research.*

Отримано: 31.08.2024