

## ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

УДК 004.4

Т. Б. Мартинюк, О. В. Войцеховська, О. С. Городецька, А. К. Рижков

## МОДУЛЬ ІНТЕГРАЦІЇ ВЕБЗАСТОСУНКІВ ІЗ ШТУЧНИМ ІНТЕЛЕКТОМ

Вінницький національний технічний університет, Вінниця

**Анотація.** В роботі досконалено метод інтеграції серверної частини вебзастосунок для організації дієти та системи тренувань зі штучним інтелектом ChatGpt, що дало змогу покращити User Experience при використанні серверної частини вебзастосунку та збільшити швидкість обробки запитів штучним інтелектом та пришвидшити роботу користувача зі штучним інтелектом. Проаналізовано сучасні технології для зв'язку серверної частини веб-додатку із базою даних, реалізації авторизації з метою впровадження захисту серверної частини від злоумисників, а також методи інтеграції веб-застосунку з штучним інтелектом. За результатами аналізу обрано стек технологій, зокрема для розробки серверної частини використано Web-API, для зв'язку серверної частини вебзастосунку із базою даних було використано технологію Entity Framework, авторизацію забезпечено за допомогою JWT Tokens. Розроблено серверну частину веб-застосунку для організації тренувань та дієти мовою програмування C#, з урахуванням принципів об'єктно-орієнтованого програмування SOLID. Під час розробки використано багаторівневу архітектуру для розподілення різних функціональних можливостей між різними модулями серверної частини. Налаштовано зв'язок серверної частини вебзастосунку з базою даних для отримання та збереження інформації. Реалізовано механізм авторизації на JWT-Токенах, що дозволило захистити методи для інтеграції з штучним інтелектом від злоумисників та дало можливість тільки авторизованим користувачам використовувати штучний інтелект для побудови тренувань та дієти. Спроектowano базу даних MSSQL, в якій зберігаються персональні дані користувача, що використовуються для побудови дієти та системи тренувань штучним інтелектом ChatGpt. Запити до бази даних оптимізовані за рахунок використання техніки Eager Loading та запитів Deferred Execution.

**Ключові слова:** вебзастосунок, модуль інтеграції, штучний інтелект, ChatGpt, багаторівнева архітектура, .NET, Access Token.

**Abstract.** The method of integrating the server part of the web application for organizing a diet and workouts with artificial intelligence ChatGpt was improved in the work, which made it possible to improve the User Experience when using the server part of the web application and increase the speed of processing requests by artificial intelligence and speed up the user's work with artificial intelligence. Modern technologies for the connection of the server part of the web application with the database, the implementation of authorization in order to implement the protection of the server part from intruders and the methods of integrating the web application with artificial intelligence are analyzed. Based on the results of the analysis, a stack of technologies was chosen, in particular, Web-API was used to develop the server part, Entity Framework technology was used to connect the server part of the web application with the database, and authorization was provided using JWT Tokens. The server part of the web application for workouts and diet organization was developed in the C# programming language, taking into account the principles of SOLID object-oriented programming. During the development, a multi-level architecture was used to distribute various functionalities between different modules of the server part. The connection of the server part of the web application with the database for receiving and saving information is configured. An authorization mechanism on JWT-Tokens was implemented, which made it possible to protect the methods for integration with artificial intelligence from attackers and gave the opportunity to only authorized users to use artificial intelligence to build workouts and diets. An MSSQL database that stores personal user data used to build the ChatGpt AI diet and workouts is designed. Database queries are optimized by using the Eager Loading technique and Deferred Execution queries.

**Key words:** web application, integration module, artificial intelligence, ChatGpt, multi-tier architecture, .NET, Access Token.

**DOI:** <https://doi.org/10.31649/1999-9941-2024-59-1-5-12>.

## Вступ

Вебзастосунки вже давно стали невід'ємною частиною нашого життя. Вони використовуються в різних сферах, включаючи освіту, бізнес, розваги, соціальні мережі та багато інших. Широке розповсюдження вебзастосунків пояснюється їх зручністю, доступністю та гнучкістю. Вони дозволяють користувачам виконувати різноманітні завдання, не виходячи з браузера.

## Актуальність

Незважаючи на широке розповсюдження, вебзастосунки постійно потребують вдосконалення. Одним з напрямків цього вдосконалення є інтеграція з чатом GPT, яка може значно покращити функціональність вебзастосунків, додаючи можливості для більш ефективної взаємодії з користувачами.

Чат GPT може використовуватися для виконання різноманітних завдань, таких як відповідь на запитання користувачів, надання рекомендацій, автоматизація процесів та багато іншого [1]. Це може зробити вебзастосунки більш інтерактивними, зручними та корисними для користувачів. Таким чином, інтеграція вебзастосунків з чатом GPT є важливим кроком на шляху до створення більш інтелектуальних та корисних вебзастосунків.

## Постановка задачі

В даній роботі інтеграцію із чатом GPT реалізовано на прикладі розробки модулю інтеграції, що використовується у вебзастосунку для організації тренувань та дієти. Більшість спортивних сайтів та застосунків можуть запропонувати користувачам лише статті на тему харчування, тренування та розра-

хунок денної норми калорій, в залежності від даних користувача, але на сьогодні жоден із сайтів не може побудувати повноцінну персоналізовану дієту та тренувальну програму, виходячи із індивідуальної інформації, що надав користувач. Тому розробка вебзастосунку для організації планування дієти та тренувань з використанням чату GPT є актуальною та перспективною задачею.

### Мета

Метою роботи є розширення функціональних можливостей вебзастосунку завдяки його інтеграції із чатом GPT, що дасть можливість покращити User Experience (UX) та збільшити швидкість обробки запитів чатом GPT.

### Основна частина

При розробці вебзастосунку для організації тренувань та дієти було вирішено використовувати ChatGpt, оскільки штучний інтелект має змогу проаналізувати інформацію про користувача, а саме індивідуальні характеристики, його мету та медичні обмеження, і, враховуючи її, побудувати найкращий денний раціон, розрахувати калорії, вказати, які вправи повинен робити користувач для досягнення результату. При цьому необхідно налаштувати ChatGpt так, щоб він спілкувався з користувачем, наче він персональний дієтолог чи тренер, який надає рекомендації щодо дієти та тренувань.

Для розробки вебзастосунку було обрано клієнт-серверну архітектуру, оскільки клієнт-серверний розподіл допомагає забезпечити ефективну обробку запитів, підвищити масштабованість та краще керувати ресурсами.

Проведено аналіз та вибір технологій для розробки клієнтської та серверної частини вебзастосунку. Клієнтську частину розроблено із використанням технології Qwik, оскільки вона орієнтована на досягнення максимальної продуктивності та оптимізації роботи веб-застосунку та зосереджується на уникненні надмірного завантаження коду JavaScript, що призводить до швидкого завантаження сторінок та поліпшення інтерактивності для користувача [2].

При проектуванні серверної частини обрано багаторівневу архітектуру, яка розділяє рівень доступу до даних, рівень бізнес логіки та рівень представлення на окремі шари [3, 4]. Для розробки API було обрано платформу ASP.Net Core та мову програмування C#. Для взаємодії з базою даних обрано Entity Framework Core, який відповідає за зручну маніпуляцію над даними через об'єктно-орієнтований підхід платформи .Net [5]. Окрім цього потрібно забезпечити авторизацію користувачів, для того щоб захистити методи інтеграції з штучним інтелектом. Для цього було обрано авторизацію на JWT-токенах [6-9].

В роботі вдосконалено метод інтеграції серверної частини з штучним інтелектом, в якому реалізовано побудову та надсилання запитів, створених на основі персональної інформації користувача, до штучного інтелекту ChatGpt з використанням його API. UML-діаграму діяльності методу інтеграції наведено на рис. 1. Коли клієнт надсилає запит, серверна частина перевіряє чи користувач, який запитав дієту існує, і при позитивному результаті відбувається перевірка чи користувач додав усю необхідну інформацію для побудови запиту.

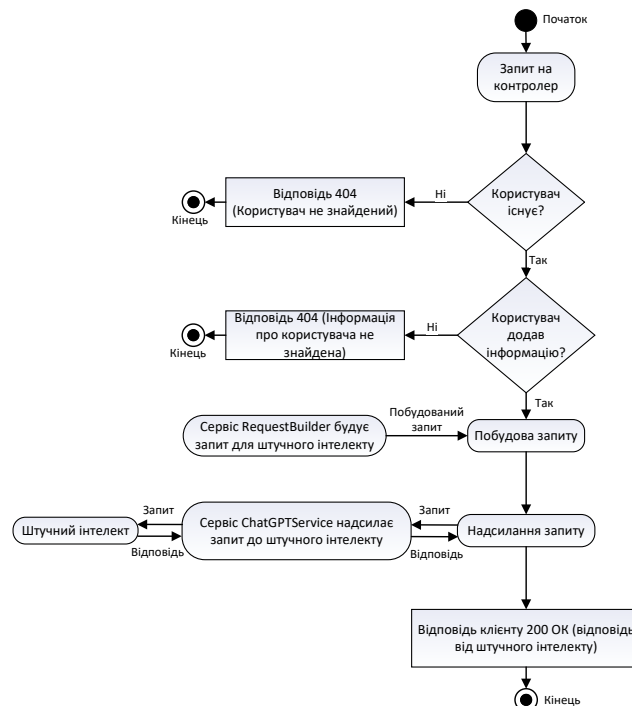


Рисунок 1 – UML-діаграма діяльності методу інтеграції серверної частини з штучним інтелектом

Якщо верифікація пройшла успішно, сервіс серверної частини буде запит до ChatGPT. При цьому, запит повинен містити усю необхідну інформацію про користувача, зокрема вік, стать, зріст та вагу. Також штучному інтелекту важливо розуміти рівень фізичної активності користувача, оскільки це напряму впливає на кількість калорій, які користувач може споживати за добу. Важливо, щоб ChatGPT розумів ціль користувача, якою може бути: втрата зайвої ваги, збереження своєї форми, набір м'язової ваги тощо. Користувач, за бажанням, може вказати свої вподобання у їжі та будь-які алергії чи непереносимість продуктів. Це допоможе штучному інтелекту сформулювати індивідуальні поради, рекомендації та плани на основі унікальних потреб користувача.

Згенерований запит надсилається до ChatGPT і серверна частина очікує на опрацювання запиту. При отриманні відповіді на запит, вона буде надіслана на клієнтську частину для відображення на рівні представлення.

Запропонований метод дозволяє заповнити просту форму на клієнтській частині, і з цієї інформації серверна частина побудує запит. В свою чергу, інші веб-застосунки, що інтегровані з штучним інтелектом, змушують користувача самому будувати запит до штучного інтелекту, через що користувач може забути вказати важливу інформацію або побудувати запит неправильно, з точки зору prompt engineering [10, 11].

У вебзастосунку використано ChatGPT версії 3.5-turbo. Приклад згенерованого серверною частиною запиту наведено на рис. 2. Запит у JSON-форматі буде надісланий на API ChatGPT.

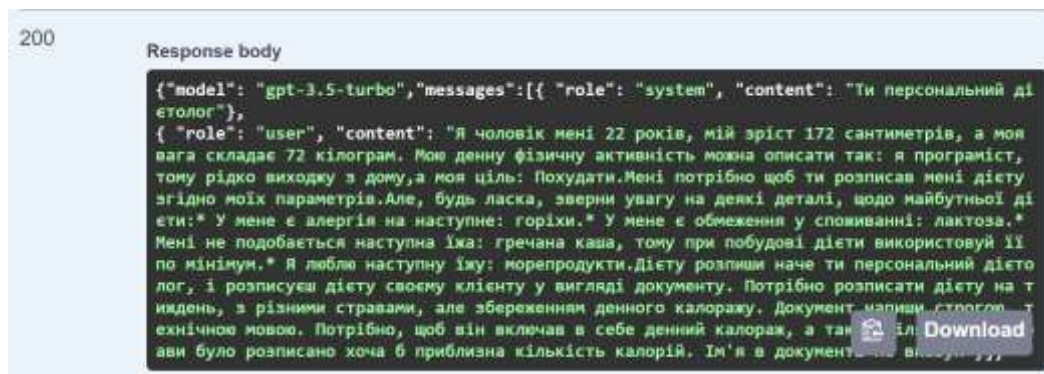


Рисунок 2 – Скріншот запиту, сформованого серверною частиною у JSON-форматі

В запиті вказано необхідну інформацію про користувача, наявність алергії та обмеження в їжі, харчові вподобання. Окрім цього вказано, що дієта повинна мати вигляд документу, в якому буде розписано денну допустиму кількість калорій для користувача, а також приблизний калораж кожної страви.

У відповідь на запит, що наведено на рис. 2, ChatGPT згенерує відповідь клієнту, в якій буде вказано раціон харчування на тиждень, кількість калорій в кожній страві, а також скільки всього калорій потрібно отримувати користувачу в день, щоб досягти своєї мети.

Аналогічно формується запит для побудови тренування для користувача, відповідно до його цілей. У цьому випадку штучний інтелект налаштовуємо на те, щоб він був персональним тренером, замість персонального дієтолога. Інформація, що вводиться, подібна до тієї, яка використовувалась при побудові запиту на дієту.

Для реалізації описаного вище методу інтеграції було розроблено програмний модуль інтеграції з штучним інтелектом, UML-діаграма класів якого наведена на рис. 3.

Модуль інтеграції вебзастосунку для організації тренувань та дієти зі штучним інтелектом складається з двох класів та двох інтерфейсів для забезпечення перевикористання та гнучкості коду:

- інтерфейс `IChatRequestBuilder`, в якому визначені публічні функції, що необхідні для отримання запиту для побудови дієти та тренувань;
- інтерфейс `IChatGptService`, в якому визначені усі публічні методи, необхідні для інтеграції зі штучним інтелектом;
- клас `ChatRequestBuilder`, в якому реалізовано методи для побудови запиту для отримання дієти та тренувань, а також приватні методи, в яких винесено базові елементи повідомлення штучному інтелекту та додавання харчових звичок відповідно до їх наявності;
- клас `ChatGptService`, в якому реалізовано метод, що ініціалізує клієнта та метод, що надсилає запит штучному інтелекту і отримує відповідь.

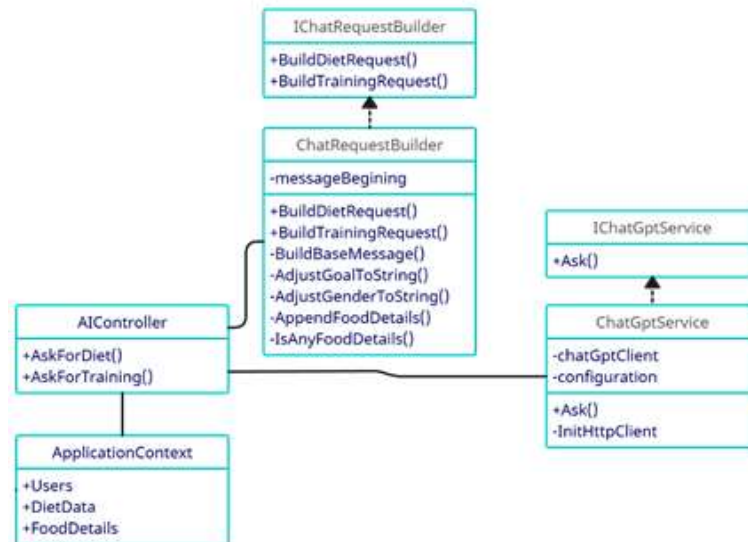


Рисунок 3 – UML-діаграма класів модулю інтеграції вебзастосунку із ChatGPT

Тригером до запуску усього процесу є контролер `AIController`, який приймає запит від клієнтської частини. Після отримання запиту контролер отримує інформацію про користувача із бази даних, передає її у клас `ChatRequestBuilder` для побудови запиту на дієту або тренування та надсилає цей запит до `ChatGpt`. Після отримання відповіді, клас контролеру відповідає на клієнтський запит та пересилає відповідь штучного інтелекту.

#### Тестування вебзастосунку

Для генерації системи тренувань та дієти користувач повинен зареєструватись та авторизуватись за електронною поштою і паролем, вказати антропометричні дані та за бажанням харчові звички, наявність алергії тощо. Зареєстрований користувач буде зберігатись в базі даних. Приклад додавання інформації, необхідної для побудови запиту наведено на рис. 4. Приклад додавання інформації про харчові звички наведено на рис. 5.

Штучний інтелект обробляє запит і у відповіді надсилає дієту або систему тренувань у вигляді тексту та деяку службову інформацію про виконання запиту (`index` – індекс повідомлення від штучного інтелекту; `message` – відповідь чату GPT у вигляді тексту; `finish_reason` – причина закінчення генерації відповіді; `usage` – об'єкт, який містить інформацію про кількість використаних токенів на генерацію відповіді). Відповідь штучного інтелекту з наданою дієтою на тиждень наведено у лістингу 1.

```
Request body
{
  "userId": "4c2514b5-f648-44fb-b4d5-0fb996d7f341",
  "age": 22,
  "height": 173,
  "weight": 69,
  "gender": 1,
  "goal": 1,
  "physicalActivity": "програміст, сидячий образ життя"
}

Code    Details
200
Response headers
access-control-allow-credentials: true
access-control-allow-origin: http://localhost:5087
content-length: 0
date: Mon, 20 Nov 2023 18:40:05 GMT
server: Kestrel
vary: Origin
```

Рисунок 4 – Приклад додавання інформації про користувача

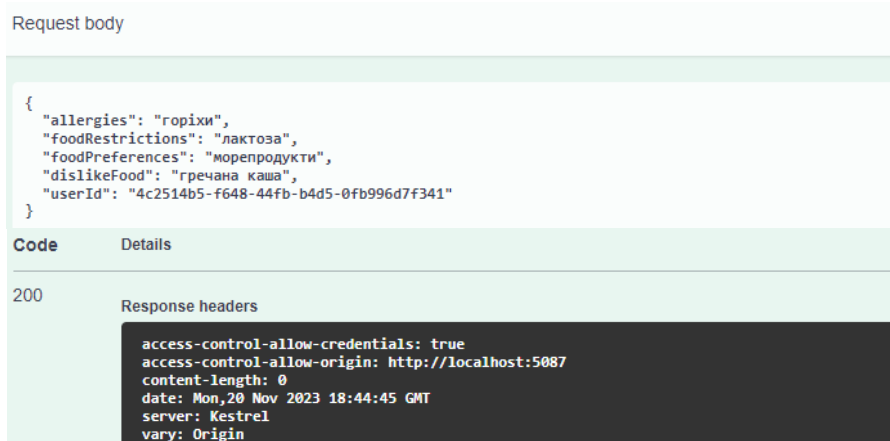


Рисунок 5 – Приклад додавання інформації про харчові звички користувача

Лістинг 1 – Відповідь штучного інтелекту на запит побудови дієти

```
{
  "id": "chatcmpl-8N3gu5xOZuLwmROAbSNaxx4GX811b",
  "object": "chat.completion",
  "created": 1700506424,
  "model": "gpt-3.5-turbo-0613",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "Дієтологічне розслідування N001\n\nЗгідно вимоги клієнта, вирішили скласти індивідуальну дієту з метою збереження маси, враховуючи її параметри та особливості. Враховуючи алергію на горіхи та обмеження щодо споживання лактози, були виключені відповідні продукти з раціону, замінені альтернативними варіантами задля забезпечення балансованого раціону. Також були враховані особисті вподобання клієнта щодо харчування і подані описи кожної страви з приблизною кількістю калорій.\n\nНагадуємо, що дана дієта розрахована на тиждень, з використанням різних страв для різноманітності, збереженням денного калоражу, а включення морепродуктів, що предпочитає клієнт, є обов'язковим.\n\n**План харчування**\n\n*День 1*\n\nСніданок: Омлет з двох яєць з додаванням цибулі та перцю - 240 ккал.\n\nСніданок №2: Нежирний йогурт з фруктами - 150 ккал.\n\nОбід: Грильована куряча грудка з овочами та рисом - 350 ккал.\n\nПолуденок: Яблуко - 80 ккал.\n\nПолуденок №2: Фруктовий салат із сезонних фруктів - 120 ккал.\n\nВечеря: Запечений лосось з пюре зі смаженими овочами - 400 ккал.\n\nПізна вечеря: Половина грейпфрута - 50 ккал.\n\n*День 2*\n\nСніданок: Нежирний йогурт з м'ясом та свіжими ягодами - 250 ккал.\n\nСніданок №2: Фруктовий смузі з молоком без лактози - 180 ккал.\n\nОбід: Тушкована свинина зі стевією та овочевим гарніром - 400 ккал.\n\nПолуденок: Груша - 60 ккал.\n\nПолуденок №2: Зелена салатна суміш з оливковою олією та орехами (без горіхів) - 150 ккал.\n\nВечеря: Печена тріска з лимона та овочевим гірчичним соусом - 350 ккал.\n\nПізна вечеря: Чашка чорного чаю - 0 ккал.\n\n*День 3*\n\nСніданок: Ржани оладки з лляним насінням та медом - 280 ккал.\n\nСніданок №2: М'яка моцарелла з томатами та базиліком - 200 ккал.\n\nОбід: Варений індича філе з квасолею та салатом зі свіжих овочів - 380 ккал.\n\nПолуденок: Мандарин - 40 ккал.\n\nПолуденок №2: Ароматизований йогурт без лактози - 120 ккал.\n\nВечеря: Гречані локшина з креветками та соєвим соусом - 450 ккал.\n\nПізна вечеря: Один шматочок чорного шоколаду (70% какао) - 60 ккал.\n\nІ так далі, розписуючи план харчування на тиждень із різними стравами, але зберігаючи денний калораж клієнту.\n\nЦей план харчування розроблено з урахуванням денного калоражу клієнта та її потреби в збереженні маси. Приблизна кількість калорій наведена для кожної страви, щоб забезпечити належний контроль над харчуванням.\n\nСупроводжуючи цей план харчування, радимо щодня споживати достатню кількість води, не менше 1,5-2 літрів, і займатися фізичною активністю для підтримання загальної фізичної форми.\n\nЦей план харчування підлягає регулярному оновленню та адаптації, в залежності від реакції клієнта та досягнення її цілей.\n\nПідпис: персональний дієтолог"
```

```
    },
    "finish_reason": "stop"
  ]
},
"usage": {
  "prompt_tokens": 588,
  "completion_tokens": 1559,
  "total_tokens": 2147
}
```

У дієті враховані обмеження користувача та його вподобання. Окрім цього кожна страва містить кількість калорій, необхідних користувачу. Видно, що штучний інтелект повертає текст із символами форматування, тому на рівні представлення розробники можуть відобразити дієту згідно UI/UX дизайну.

Аналогічно штучний інтелект надав програму тренувань для користувача. Відповідь штучного інтелекту з розписаною системою тренувань наведено у лістингу 2.

Лістинг 2 – Відповідь штучного інтелекту на запит побудови системи тренувань

```
{
  "id": "chatcmpl-8N3rxSLkup3ArbwYFf4VGj1AtKSyz",
  "object": "chat.completion",
  "created": 1700507109,
  "model": "gpt-3.5-turbo-0613",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "Розклад тренувань для досягнення цілі -- збереження маси тіла:\n\nДень 1:
Нижня частина тіла\n\n1. Розминка (10 хвилин):\n - Швидка хода на біговій доріжці (5
хвилин)\n - Розтяжка ніг (5 хвилин)\n\n2. Силові вправи (3 набори по 10-12 повторень):\n -
Присідання з гантелями\n - Промовистий жим ніг\n - Випади з гантелями\n - Мостик\n\n3.
Кардіо-тренування (20-30 хвилин):\n - Велотренажер\n - Кросовер або еліптичний
тренажер\n\n4. Розтяжка (10 хвилин):\n - Статичні розтяжки для ніг та стегон\n\nДень 2:
Верхня частина тіла\n\n1. Розминка (10 хвилин):\n - Загальне розігрівання (5 хвилин)\n -
Розтяжка рук та плечей (5 хвилин)\n\n2. Силові вправи (3 набори по 10-12 повторень):\n - Жим
штанги на груди\n - Розведення гантелей на плечі\n - Тяга гантель до плеча\n - Вправи
для трапеції\n\n3. Кардіо-тренування (20-30 хвилин):\n - Біг на біговій доріжці\n - Верти-
кальний еліптичний тренажер\n\n4. Розтяжка (10 хвилин):\n - Розтяжка м'язів спини та
рук\n\nДень 3: Загальне зміцнення\n\n1. Розминка (10 хвилин):\n - Інтервальний біг на місці
(5 хвилин)\n - Розтяжка всього тіла (5 хвилин)\n\n2. Функціональні тренування (3-4 набори по
10-12 повторень):\n - Підйоми на боковій планці\n - Планка на передню та задню дельту\n
- Русская рулетка з гантелями\n - Супермени\n\n3. Кардіо-тренування (20-30 хвилин):\n -
Тренажер для бігу зі зростанням нахилу\n - Степ-платформа\n\n4. Розтяжка (10 хвилин):\n -
Інтервал розтяжок всіх м'язових груп тіла\n\nПримітка: Після кожної тренування, не забудьте
зайти на розтяжку зону, щоб розслабити м'язи та попрацювати над гнучкістю.\n\nЗверніть увагу,
що цей розклад є загальним і може бути змінений або адаптований відповідно до вашої фізичної
підготовки та ваших потреб. Рекомендую також проконсультуватися з особистим тренером, який
допоможе вам дещо продумати, зокрема, раціон харчування, тренувальний план тощо."
      },
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 256,
    "completion_tokens": 1078,
    "total_tokens": 1334
  }
}
```

Отримана система тренувань описує які вправи слід виконувати та в які дні. Також описано загальні рекомендації щодо занять спортом, щоб користувач міг коригувати систему тренувань під себе.

### Висновки

1. Вдосконалено метод інтеграції веб-застосунку з штучним інтелектом ChatGPT 3.5-turbo за рахунок побудови та надсилання запиту до OpenAI API, сформованого на основі визначених в роботі вимог, що дало можливість покращити User Experience при використанні веб-застосунку.

2. Спроектовано багаторівневу архітектуру серверної частини веб-застосунку, що дозволило розділити різні функціональні можливості у різних модулях. Програмна реалізація серверної частини веб-застосунку для організації тренувань та дієти виконана на платформі ASP.Net Web-Api, з дотриманням принципів об'єктно-орієнтованого програмування та принципів SOLID.

3. Реалізовано механізм авторизації на JWT-Токенах, що дозволило захистити методи для інтеграції з штучним інтелектом від злоумисників та дало можливість тільки авторизованим користувачам використовувати штучний інтелект для побудови тренувань та дієти.

4. Проведене тестування показало відсутність критичних помилок, стабільність роботи клієнтської і серверної частин та правильність виконання заданих функцій.

### Список літератури

- [1] Introducing ChatGPT and Whisper APIs. [Електронний ресурс]. Режим доступу: <https://openai.com/blog/introducing-chatgpt-and-whisper-apis>.
- [2] Фреймворк Qwik. [Електронний ресурс]. Режим доступу: <https://qwik.builder.io/docs/concepts/think-qwik/>.
- [3] Understanding Multilayer Architecture. [Електронний ресурс]. Режим доступу: <https://www.c-sharpcorner.com/UploadFile/1492b1/understanding-multilayered-architecture-in-net/>
- [4] A Multi-Layer Back-End Application Architecture in .NET Core. [Електронний ресурс]. Режим доступу: <https://hamzaak.medium.com/a-multi-layer-back-end-application-architecture-in-net-core-c08898f2427e>
- [5] Introduction to Entity Framework. [Електронний ресурс]. Режим доступу: <https://www.partech.nl/nl/publicaties/2020/11/introduction-to-entity-framework>
- [6] Войцеховська О. В., Городецька О. С., та Рижков А. К., “Аналіз методів авторизації при проектуванні серверної частини веб-додатку” на Матеріали міжнародної науково-практичної інтернет-конференції “Електронні інформаційні ресурси: створення, використання, доступ”, 2023 р. С. 225–227. Режим доступу: [https://drive.google.com/file/d/1oVmxS3W\\_sEQPjes9S9AzWDaJxDxi6I0X/view](https://drive.google.com/file/d/1oVmxS3W_sEQPjes9S9AzWDaJxDxi6I0X/view).
- [7] Authentication and Authorization in Web API. [Електронний ресурс]. Режим доступу: <https://dotnettutorials.net/lesson/authentication-and-authorization-in-web-api/>
- [8] Introduction to authorization in ASP.NET Core. [Електронний ресурс]. Режим доступу: <https://learn.microsoft.com/en-us/aspnet/core/security/authorization/introduction?view=aspnetcore-7.0>
- [9] JSON Web Tokens. [Електронний ресурс]. Режим доступу: <https://auth0.com/docs/secure/tokens/json-web-tokens>
- [10] Prompt engineering. [Електронний ресурс]. Режим доступу: <https://platform.openai.com/docs/guides/prompt-engineering>.
- [11] Embeddings – OpenAI API. [Електронний ресурс]. Режим доступу: <https://platform.openai.com/docs/guides/embeddings>.

Стаття надійшла: 12.01.2024

### References

- [1] Introducing ChatGPT and Whisper APIs. [Online]. Available: <https://openai.com/blog/introducing-chatgpt-and-whisper-apis>.
- [2] Framework Qwik. [Online]. Available: <https://qwik.builder.io/docs/concepts/think-qwik/>. Accessed on: 07. 01. 2024.
- [3] Understanding Multilayer Architecture. [Online]. Available: <https://www.c-sharpcorner.com/UploadFile/1492b1/understanding-multilayered-architecture-in-net/>. Accessed on: 07. 01. 2024.
- [4] A Multi-Layer Back-End Application Architecture in .NET Core. [Online]. Available: <https://hamzaak.medium.com/a-multi-layer-back-end-application-architecture-in-net-core-c08898f2427e>. Accessed on: 07. 01. 2024.
- [5] Introduction to Entity Framework. [Online]. Available: <https://www.partech.nl/nl/publicaties/2020/11/introduction-to-entity-framework>. Accessed on: 07. 01. 2024.
- [6] Voitsekhovska O. V., Horodetska O.S., Ryzhkov A. K., “Analiz metodiv avtoryzatsii pry proektuvanni servernoi chastyny veb-dodatku [Analysis of authorization methods when designing the server part of a web application]” on Materials of the international scientific and practical Internet conference: stvorennia, vykorystannia, dustup [Electronic information resources: creation, use, access]”, 2023. P. 225–227 [Online].
- [7] Authentication and Authorization in Web API. [Online]. Available: <https://dotnettutorials.net/lesson/authentication-and-authorization-in-web-api/>. Accessed on: 07. 01. 2024.
- [8] Introduction to authorization in ASP.NET Core. [Online]. Available: <https://learn.microsoft.com/en-us/aspnet/core/security/authorization/introduction?view=aspnetcore-7.0>. Accessed on: 07. 01. 2024.
- [9] JSON Web Tokens. [Online]. Available: <https://auth0.com/docs/secure/tokens/json-web-tokens>. Accessed on: 07. 01. 2024.
- [10] Prompt engineering. [Online]. Available: <https://platform.openai.com/docs/guides/prompt-engineering>. Accessed on: 07. 01. 2024.
- [11] Embeddings – OpenAI API. [Online]. Available: <https://platform.openai.com/docs/guides/embeddings>. Accessed on: 07. 01. 2024.

**Відомості про авторів**

**Мартинюк Тетяна Борисівна** – д.т.н., професор, професор кафедри обчислювальної техніки.

**Войцеховська Олена Валеріївна** – к.т.н., доцент, доцент кафедри обчислювальної техніки.

**Городецька Оксана Степанівна** – к.т.н., доцент, доцент кафедри обчислювальної техніки.

**Рижков Андрій Костянтинович** – студент групи ІКІ-22м кафедри обчислювальної техніки, факультет інформаційних технологій та комп'ютерної інженерії.

T. B. Martyniuk, O. V. Voytsekhovska, O. S. Horodetska, A. K. Ryzhkov

**INTEGRATION MODULE OF WEB APPLICATIONS WITH  
ARTIFICIAL INTELLIGENCE**

Vinnitsia National Technical University, Vinnitsia