

УДК 004.021

РОЗРОБКА ТА ВПРОВАДЖЕННЯ WEB-ЗАСТОСУНКУ ОЦІНКИ СКЛАДНОСТІ АЛГОРИТМІВ

Безгодков С. Р.

Науковий керівник – к.т.н., доцент Міщеряков Ю. В.

Харківський національний університет радіоелектроніки, каф. СТ

м. Харків, Україна

e-mail: sevastian.bezghodkov@nure.ua

The project involves the development and implementation of a web-based application for assessing algorithm complexity. This application will consist of both frontend and backend components, enabling users to analyze the complexity of various algorithms through a user-friendly interface. By providing tools for assessing algorithmic complexity, the application aims to support developers in optimizing their code and improving overall system performance.

Алгоритми в інформаційних технологіях є основою для розв'язання різноманітних завдань, від пошуку оптимального шляху у мережі до обробки великих обсягів даних. Вони допомагають зрозуміти, які кроки потрібно виконати для досягнення певної мети, забезпечити швидке та ефективне виконання програм, оптимізувати роботу систем і покращити їхню продуктивність. Алгоритми є ключовим елементом у всіх галузях ІТ, від розробки програмного забезпечення до штучного інтелекту, і вони необхідні для розвитку інноваційних технологій та вирішення складних завдань у цифровому світі.

Однією з ключових характеристик алгоритмів в ІТ є їх складність, яка визначається кількістю операцій, потрібних для виконання, та обсягом ресурсів, які вони вимагають. Складні алгоритми можуть потребувати значних обчислювальних потужностей та часу для виконання, що може призвести до затримок у відповідях системи або навіть неприйнятних строків виконання завдань. Тому одним із завдань розробників є розробка ефективних алгоритмів, які забезпечують швидке виконання задач і економію ресурсів.

Обчислення складності алгоритмів важливо з двох основних причин. По-перше, воно дозволяє передбачити та уникнути можливих проблем, пов'язаних з обсягом обчислювальних ресурсів та часом виконання. Знання складності алгоритмів допомагає розробникам правильно оцінювати його придатність для конкретного завдання та вибрати найефективніші рішення. По-друге, покращення складності алгоритмів важливо для оптимізації роботи програм і систем в цілому. Шляхи покращення можуть включати вдосконалення алгоритмів для зменшення кількості операцій або використання більш ефективних алгоритмічних підходів, що в результаті забезпечує підвищення продуктивності та швидкості виконання програмного забезпечення.

Вимоги, шляхи досягнення мети.

У доповіді розглядаються етапи проектування та розробки системи застосунку для покращення алгоритмів. Для розробленої системи використовується API Chat GPT. API потрібен щоб користувачеві був запропонований кращий алгоритм.

Особлива увага при проектуванні системи була приділена забезпеченню ефективного використання API, а також стабільності та безпеки зв'язку з ним. Інтеграція з API Chat GPT дозволяє системі оперативно отримувати доступ до сучасних моделей машинного навчання, що в свою чергу сприяє постійному покращенню якості алгоритмів та взаємодії з користувачем.

Задля правильного вираховування алгоритму [5] було проведено детальний аналіз потреб користувачів та їх взаємодії з системою. На основі цього аналізу були визначені основні критерії ефективності та точності, які повинен відповідати обраний алгоритм для досягнення найкращого результату в роботі з користувачем.

В якості СУБД [3] обрано MySQL [2] з використанням реляційної бази даних. Окрім наявності досвіду використання даної СУБД, вона має ще ряд інших переваг.

MySQL – це одна з найпопулярніших відкритих реляційних баз даних. Реляційні бази даних [3] використовують реляційну модель даних, яка забезпечує ефективне зберігання і обробку даних. Реляційна модель даних використовує таблиці, які складаються з рядків і стовпців.

Для реалізації сервісної частини системи створено проєкт Web API. Обрано мову програмування C# [4].

Список використаних джерел:

1. Mullins C. S. DEFINITION database management system (DBMS) [Електронний ресурс] / Craig S. Mullins. – 2023. – Режим доступу до ресурсу: <https://www.techtarget.com/searchdatamanagement/definition/database-management-system>.

2. MySQL Documentation [Електронний ресурс]. – 1995. – Режим доступу до ресурсу: <https://dev.mysql.com/doc/>.

3. Реляційні бази даних усе, що необхідно про них знати [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://foxminded.ua/reliatsiini-bazy-danykh/>.

4. .Net Documentation [Електронний ресурс]. – 1991. – Режим доступу до ресурсу: <https://learn.microsoft.com/dotnet/csharp/>.

5. Algorithms documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.linkedin.com/advice/0/how-can-you-use-algorithm-documentation-communicate>.