

УДК 004.422:378.018.43

РОЗРОБКА КОМПОНЕНТІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПЕРЕВІРКИ НАВЧАЛЬНИХ ЗАВДАНЬ З АВТОМАТИЗОВАНОЮ ОЦІНКОЮ КОДУ

Глова С.О., Хряпкін О.В.

e-mail: stanislav.hlova@nure.ua, oleksandr.khriapkin@nure.ua

Харківський національний університет радіоелектроніки, каф. СТ
м. Харків, Україна

This work focuses on the development of an automated system for assessing student programming assignments using GitHub Organizations and Jenkins. The system utilizes template repositories, private student repositories, and hidden tests to ensure objective evaluation. Integration with Moodle via Learning Tools Interoperability (LTI) enables seamless adoption in Learning Management System (LMS) environments. The proposed approach enhances transparency, reduces grading time, and provides instant feedback to students.

У доповіді розглядається система, яка забезпечує прозоре оцінювання студентського коду, скорочення часу на перевірку та миттєвий зворотний зв'язок для студентів, що є важливим в умовах дистанційного навчання. Автоматизація цього процесу забезпечує ефективний механізм оцінювання програмних робіт, зменшуючи вплив людського фактора та забезпечуючи прозорість процесу перевірки. Для автоматизації перевірки студентських навчальних робіт з автоматизованою оцінкою коду використовуються GitHub Organizations для централізованого управління студентськими репозиторіями та Jenkins для реалізації механізмів автоматичного оцінювання [1,2].

Система підтримує дві основні ролі користувачів: лектор та студент. Лектор створює завдання шляхом створення окремого шаблонного репозиторію в GitHub Organization. Шаблонний репозиторій – це спеціальний тип репозиторію, який використовується для створення однакових за структурою репозиторіїв без копіювання історії комітів. Це дозволяє забезпечити стандартизований стартовий код для всіх студентів, уникнувши проблем із форкуванням, що могло б призвести до порушення конфіденційності прихованих тестів. Приховані тести – окремі тестові сценарії, які не містяться в репозиторіях студентів і додаються лише на етапі перевірки. Використання прихованих тестів є важливим елементом контролю коректності студентських рішень. Вони дозволяють перевіряти програму на непередбачувані сценарії, які студент не може підлаштувати під тестові дані, що забезпечує об'єктивність оцінювання.

При створенні завдання лектор визначає параметри перевірки, зокрема:

- мову програмування, версію середовища виконання, менеджер збірки;
- правила оцінювання (розподіл балів за успішну збірку та проходження тестів);

– кількість дозволених спроб та дедлайни.

Студент виконує завдання у власному приватному репозиторії, який автоматично створюється в GitHub Organization після підтвердження участі у завданні. При цьому студент отримує доступ лише до відкритої частини шаблонного репозиторію без прихованих тестів. Використання приватного репозиторію гарантує, що студент не має доступу до чужих рішень і не може отримати приховані тести до моменту перевірки. Це забезпечує індивідуальність виконання завдань і виключає можливість плагіату або неконтрольованого обміну кодом. Час існування репозиторія студента вказується викладачем, керуючись особливостями учбового процесу.

Перевірка завдання здійснюється лише після натискання студентом визначеної для цього кнопки в графічному інтерфейсі застосунка, який інтегрується в завдання створене в LMS-системі. Після натискання створюється HTTP-запит до бекенд-серверу, який ініціює процес перевірки на Jenkins. У свою чергу Jenkins завантажує код студента, додає приховані тести, виконує збірку та тестування, після чого система аналізує результати [3]. Оцінка формується на основі успішності проходження тестів, а студент отримує детальний розподіл балів із можливістю перегляду причин помилок.

Однією з ключових особливостей запропонованого рішення є інтеграція з Moodle через LTI (Learning Tools Interoperability) [4]. Це дозволяє безпосередньо використовувати систему в реальних LMS-платформах, спрощуючи процес реєстрації студентів та викладачів у системі, а також забезпечуючи автоматичну передачу оцінок у навчальне середовище. Таким чином, запропоноване рішення може бути легко адаптоване до потреб освітніх закладів.

Архітектура системи побудована за мікросервісним підходом, що забезпечує її масштабованість, гнучкість та розширюваність. Взаємодія між компонентами здійснюється за допомогою REST API. Серверна частина системи реалізована з використанням Java та Spring Boot, що забезпечує ефективну обробку запитів, управління користувачами та взаємодію з GitHub API та Jenkins API. Дані зберігаються у реляційній базі даних MySQL. Клієнтська частина створена за допомогою React та TypeScript, що забезпечує зручний інтерфейс користувача для викладачів та студентів. Для керування студентськими репозиторіями використовується GitHub API, що дозволяє автоматично створювати приватні репозиторії студентів у GitHub Organization, а також контролювати доступ до них. Процес перевірки коду реалізований за допомогою Jenkins, який виконує автоматичне тестування студентських рішень, включаючи збірку, запуск тестів. Аналіз результатів відбувається на окремому Spring-сервісі. Використання Docker дозволяє створювати ізольовані середовища виконання, що забезпечує однакові умови перевірки

незалежно від локального середовища розробки студентів. Архітектура системи наведено на рисунку 1.

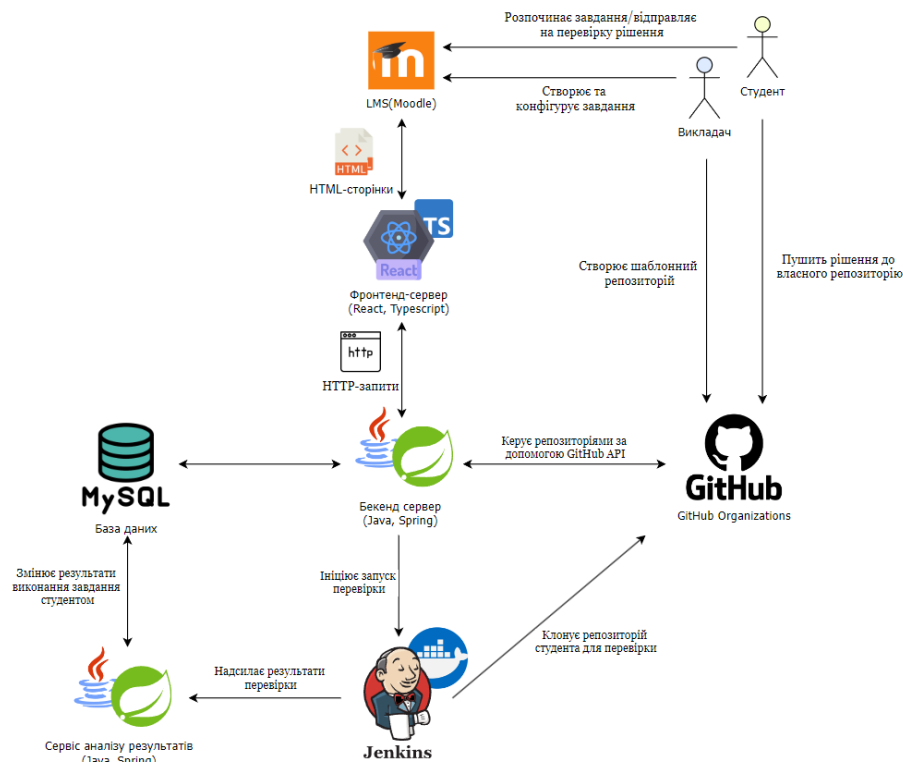


Рисунок 1 – Схема архітектури веб-застосунку

Розроблена система забезпечує підвищення ефективності перевірки студентського коду, спрощує процес оцінювання та створює єдину платформу для взаємодії студентів та викладачів у процесі навчання. Впровадження системи дозволяє значно скоротити витрати часу на перевірку робіт, підвищити прозорість оцінювання та покращити якість зворотного зв'язку.

Список використаних джерел:

1. About organizations : вебсайт. URL: <https://docs.github.com/en/organizations/collaborating-with-groups-in-organizations/about-organizations> (дата звернення: 01.03.2025).
2. Jenkins Documentation : вебсайт. URL: <https://www.jenkins.io/doc/> (дата звернення: 02.03.2025).
3. Хряпкін О. В., Безугла Г. Є. Розробка інфраструктури WEB-компонентів інформаційної системи // International science conference on multidisciplinary research, м. Берлін, 2021. С. 1113.
4. LTI and Moodle : вебсайт. URL: https://docs.moodle.org/405/en/LTI_and_Moodle (дата звернення: 03.03.2025).