

## **Розробка серверної частини системи тестування студентів**

Вінницький національний технічний університет

### **Анотація**

*В роботі досліджено серверну частину системи тестування студентів, як системи для ефективного контролю знань студентів. За допомогою такої системи викладач має можливість самостійно створювати тести зі свого предмету, а студент може самостійно перевіряти свої знання.*

**Ключові слова:** серверна частина, backend, система тестування, розробка.

### **Annotation**

*The server part of the student testing system is investigated as a system for effective control of students' knowledge. Each teacher with this system can independently create tests on their subject, and the student can independently test their knowledge.*

**Keywords:** server side, backend, testing system, development.

### **Вступ**

Сьогодні при проведенні тестування знань студентів у вигляді опитувальних листів використовується занадто багато ресурсів, як фінансових, так і трудових з боку викладача (пошук відповідних тестів, їх перевірка, виставлення оцінок тощо). Тому, цілком актуальним є бажання створити електронну систему, за допомогою якої будь-який викладач може самостійно створювати тести зі своєї дисципліни, а студент, зі свого боку, має можливість самостійно перевіряти свої знання на тестах, які запропоновані викладачем, та є у відкритому доступі для всіх бажаючих.

Метою роботи є аналіз існуючих методик тестування знань студентів та розробка оригінальної ефективної електронної системи тестування знань із зручним користувацьким інтерфейсом.

### **Розробка системи тестування студентів**

Система тестування[1] - це безкоштовна система для створення тестів і проведення тестування, призначена для викладачів і студентів будь-яких навчальних закладів. Основна мета створення цієї системи - полегшити роботу викладачам і надати студентам можливість самостійно перевірити свої знання.

Тестування використовується, як засіб об'єктивного контролю знань навчальної програми в студентів. Систематичний контроль рівня знань студентів з одного боку визначає успішність студента, а з іншого боку є показником ефективності методики навчання і організації навчального процесу. Включення різних форм тестових завдань в процесі навчання мотивує студентів до активізації роботи по засвоєнню навчального матеріалу і формує прагнення розвивати свої здібності.

При тестуванні знань студентів у вигляді опитувальних листів використовується занадто багато ресурсів, як фінансових, так і трудових з боку викладача (пошук відповідних тестів, їх перевірка, оцінка і т.д.). Кожен викладач за допомогою запропонованої системи може самостійно створювати тести зі свого предмета, а студент може самостійно перевірити свої знання на тестах, які запропоновані викладачем та відкриті для всіх.

В системі реалізована можливість обміну тестами, тобто викладачі можуть дозволити користуватися своїми тестами іншим викладачам, зареєстрованих в системі. Таким чином відбувається обмін інформацією, а також економиться час на створення тестів.

Використовуючи систему тестування, отримуємо:

- безкоштовну систему по створенню тестів та проведення тестування;
- систему, яку легко, швидко і зручно використовувати в роботі;
- необмежену кількість користувачів — викладачів і студентів;

- необмежену кількість запитань і тестів.

Якщо порівняти процедуру проведення тестування у вигляді опитувальних листів і комп'ютерне, то очевидно, що комп'ютерне тестування більш ефективно.

Отже, розробка системи для комп'ютерного тестування, що знижує витрати як фінансові, так і трудові покладені на викладача, є актуальним і ефективним рішенням.

Розглянемо технології, які використовуються для розробки:

- Мова програмування[2] (англ. Programming language) — це штучна мова, яка створена для передачі команд машинам, зокрема комп'ютерам. Мови програмування використовуються для створення програм, які контролюють поведінку машин, та запису алгоритмів. Більш строгі визначення мови програмування — це система позначень для опису алгоритмів і структур даних, певна штучна формальна система, засобами якої можна виражати алгоритми. Мову програмування визначає набір лексичних, синтаксичних і семантичних правил, що задають зовнішній вигляд програми та дії, що виконує виконавець (комп'ютер) під її управлінням.
- Фреймворк[3] (англ. Framework, каркас) - інфраструктура програмних рішень, що полегшує розробку складних систем. Спрощено дану інфраструктуру можна вважати своєрідною комплексною бібліотекою, але при цьому вона має ряд обмежень, що задають правила створення структури проекту та написання коду.
- Шаблони проектування програмного забезпечення[4] (англ. software design patterns) — ефектні способи вирішення задач проектування програмного забезпечення. Шаблон не є закінченим зразком, який можна безпосередньо транслювати в програмний код. Об'єктно-орієнтований шаблон найчастіше є зразком вирішення проблеми і відображує відношення між класами та об'єктами, без вказівки на те, як буде зрештою реалізоване це відношення.
- Сервер[5] (англ. server — «служка», від англ. to serve — служити, множ. сервери) — у комп'ютерній термінології термін може стосуватися окремого комп'ютера чи програми. Головною ознакою в обох випадках є здатність машини чи програми переважно кількість часу працювати автономно, без втручання людини, реагуючи на зовнішні події відповідно до встановленого програмного забезпечення. Втручання людини відбувається під час встановлення серверу та під час його сервісного обслуговування. Часто це роблять окремі адміністратори серверів з вищою кваліфікацією.

### Висновки

Розроблена та досліджена серверна частина із зручним користувацьким інтерфейсом електронної системи тестування студентів, як системи ефективного контролю знань студентів, за допомогою якої, будь-який викладач може самостійно створювати тести зі своєї дисципліни, а студент, зі свого боку, має можливість самостійно перевіряти свої знання на тестах, які запропоновані викладачем, та є у відкритому доступі для всіх бажаючих. Завдяки такій системі електронного тестування у викладача відсутня необхідність перевіряти виконані тести, оскільки це зробить сама система одразу після завершення тестування. Крім того, викладач має можливість спостерігати на моніторі свого комп'ютера за результатами тестування та визначити, скільки завдань виконав кожний студент, на які питання відповів, чи правильно відповів і скільки балів за це отримав, що, безумовно, суттєво полегшує працю викладача та нівелює вірогідність неадекватного оцінювання знань.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. <http://www.znanius.com/5520.html>
2. [https://uk.wikipedia.org/wiki/Мова\\_програмування](https://uk.wikipedia.org/wiki/Мова_програмування)
3. [https://uk.wikipedia.org/wiki/Програмний\\_каркас](https://uk.wikipedia.org/wiki/Програмний_каркас)
4. <https://echo.lviv.ua/dev/5432>
5. <https://uk.wikipedia.org/wiki/Сервер>

**Копиця Вадим Олександрович** – студент групи ІСТ-176, факультету комп'ютерних систем та автоматики, Вінницький національний технічний університет, м. Вінниця, kopivad@gmail.com

Науковий керівник: **Васюра Анатолій Степанович** – професор кафедри автоматизації та інтелектуальних інформаційних систем, Вінницький національний технічний університет, м. Вінниця.

***Kopitsya Vadym Oleksandrovich*** – student of group 1IST-17b, Faculty of Computer Systems and Automation, Vinnitsa National Technical University, metro Vinnitsya, kopivad@gmail.com

Supervisor: ***Vasyura Anatoly Stepanovich*** – professor at the Department of Atomization and Intelligent Information Systems, Vinnytsia National Technical University, Vinnytsia.