

Наkutest: Переосмысление образовательного тестирования

И. М. Шелепугин

Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)

shelepuginivanm@gmail.com

Аннотация. Проанализированы особенности, преимущества и недостатки существующих систем образовательного тестирования. Разработана система автоматизированного образовательного тестирования на языке программирования Go, лишённая недостатков, выявленных у существующих решений.

Ключевые слова: информатика; программирование; Go; образование; тестирование.

I. ВВЕДЕНИЕ

Тестирование признается одним из самых эффективных средств оценивания знаний. Использование тестовых заданий и их совокупностей (тестов) позволяет преподавателю объективно оценить уровень знаний ученика. К достоинствам обозначенного метода относят развитие дисциплины, повышение степени сознательности учащихся, выявление пробелов в знаниях [1].

В современной системе образования широкое распространение получило компьютерное тестирование. Оно осуществляется при помощи ПО, которое автоматизирует проверку ответов, повышает удобство тестирования. Сегодня существует большое количество систем для образовательного тестирования. Однако, зачастую они обладают рядом недостатков. У некоторых есть уязвимости в безопасности, позволяющие скомпрометировать правильные ответы или подделать результат. Некоторые из них проприетарны: данные хранятся на сторонних серверах и могут использоваться третьими лицами.

Целью работы является разработка системы для автоматизированного образовательного тестирования, в которой учитываются преимущества и недостатки существующих решений.

II. АНАЛИЗ ОСОБЕННОСТЕЙ СУЩЕСТВУЮЩИХ СИСТЕМ

В табл. 1 приведена сравнительная характеристика некоторых популярных решений, используемых для тестирования в сфере образования. Скорость работы проанализирована с помощью PageSpeed Insights [2].

ТАБЛИЦА 1.

Название	Исходный код	Назначение	Скорость работы
Google Формы	Закрытый	Общее	60/100
Яндекс Формы	Закрытый	Общее	60/100
Moodle	Открытый	Образование	64/100
NetTest	Закрытый	Образование	100/100

Таким образом, подавляющее большинство программ для массового тестирования не предоставляют исходный код, что снижает конфиденциальность пользователей [3]. Это особенно критично в сфере образования, так как данные государственных учреждений могут попасть в руки третьим лицам. Кроме того, часто для проведения сетевого тестирования учащихся применяются программы общего назначения, что является признаком нехватки специализированного ПО на рынке. Наконец, наиболее популярные сервисы показывают сравнительно низкую производительность, что негативно сказывается на пользовательском опыте [4].

Стоит отметить, что специализированные решения для тестирования зачастую обладают уязвимостями, позволяющими скомпрометировать ответы или подделать результаты. Так, NetTest хранит ответы в HTML-коде страницы, а для Moodle существует браузерное расширение [5], позволяющее видеть ответы других пользователей. Это значительно снижает общую безопасность данных систем.

III. РАЗРАБОТКА СИСТЕМЫ НАKUTEST ДЛЯ АВТОМАТИЗИРОВАННОГО ОБРАЗОВАТЕЛЬНОГО ТЕСТИРОВАНИЯ

A. Принцип работы и архитектура ПО

В основе Nakutest лежат следующие принципы:

- ПО состоит из серверного и клиентского приложений, которые взаимодействуют по сети;
- серверное приложение запускается локально, как правило, на устройстве преподавателя;
- клиентское приложение открывается в браузере на устройствах учащихся;
- данные пользователей, такие как тесты, ответы, статистика, настройки и проч. хранятся локально на устройстве.

Данные принципы позволяют системе избежать наличия недостатков, присущих её аналогам.

С точки зрения архитектуры Nakutest представляет собой веб-сервер и клиентское приложение, собранное в один исполняемый файл. Запущенная программа отображается в виде иконки в системном лотке [6]. Эти особенности упрощают интеграцию ПО; в отличие от прочих систем образовательного тестирования, Nakutest не требует сложной настройки (но предоставляет такие возможности для продвинутых пользователей).

В. Создание теста. Добавление заданий

Создание тестов в Nakutest осуществляется через веб-интерфейс программы.

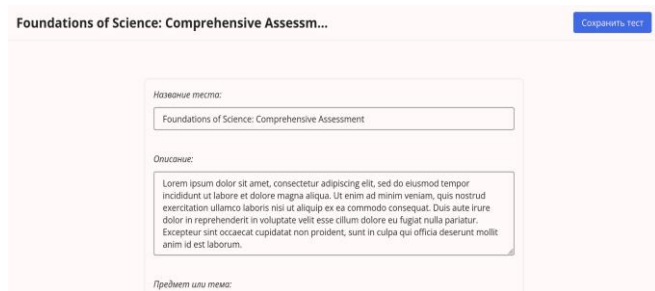


Рис. 1. Интерфейс редактирования теста

На данный момент система реализует 4 типа заданий:

- один вариант ответа;
- несколько вариантов ответа;
- открытый ответ;
- развёрнутый ответ.

Программа позволяет задать правильные ответы на первые три типа вопросов, автоматизируя проверку тестов. Развернутые ответы проверяются преподавателем вручную.

Текст задания набирается при помощи языка разметки Markdown [7], позволяющий добавлять ссылки, изображения, таблицы и проч.

Ответы на задания хранятся только на сервере и не отправляются на клиентское приложение. Это предотвращает возможность компрометации ответов и подделки результатов тестирования.

Тесты хранятся на устройстве в формате JSON. Подобная стратегия упрощает

С. Многопоточное программирование

Nakutest использует многопоточное программирование для повышения производительности. Программа написана на языке программирования Go, стандартная библиотека которого реализует структуры и функции для конкурентной разработки [8][9].

Каждый запрос, отправляемый на сервер, обрабатывается в отдельном потоке, что позволяет проверять множество тестов одновременно без существенной потери производительности.

Д. Безопасность. Политики доступа

Неотъемлемой частью любого клиент-серверного приложения является информационная безопасность. Модель безопасности Nakutest основана на политиках доступа к ресурсам. В зависимости от контекста, система обрабатывает или блокирует запрос. На текущий момент реализована поддержка следующих контекстов:

- для всех без верификации – все запросы пропускаются без дополнительной проверки;
- по имени пользователя и паролю – требует ввода учетных данных, предварительно добавленных в базу данных;

- только это устройство – доступ к маршруту возможен только с устройства, на котором запущен сервер;
- прерывать все запросы – все запросы блокируются.

Таким образом, модель безопасности Nakutest позволяет гибко регулировать доступ к ресурсам, адаптируя систему под различные сценарии использования.

Е. Документация к проекту

Документация – важная часть проекта, описывающая процессы запуска и настройки, сценарии использования. Для системы Nakutest была написана и оформлена документация [10].

Веб-сайт документации разработан с помощью генератора статических сайтов VitePress [11]. Содержание документации написано на языке разметки Markdown [7].

IV. ЗАКЛЮЧЕНИЕ

Проведен сравнительный анализ существующих систем автоматизированного образовательного тестирования. Разработана и протестирована система Nakutest. Применены паттерны многопоточного программирования. Написана и оформлена документация к проекту.

СПИСОК ЛИТЕРАТУРЫ

- [1] Задаев С.А. Эффективность тестирования при оценивании результатов обучения / С.А. Задаев, М.М. Клименко. Текст: непосредственный // Молодой ученый. 2021. № 19 (361). С. 287-289. — URL: <https://moluch.ru/archive/361/80758/> (дата обращения: 03.03.2025).
- [2] PageSpeed Insights. Текст: электронный // PageSpeed Insights : [Электронный ресурс]. — URL: <https://pagespeed.web.dev/> (дата обращения: 03.03.2025).
- [3] Satyabrata, Jena. Difference between Open Source Software and Closed Source Software / Jena Satyabrata. Текст : электронный // GeeksforGeeks: [Электронный ресурс]. — URL: <https://www.geeksforgeeks.org/difference-between-open-source-software-and-closed-source-software/> (дата обращения: 03.03.2025).
- [4] Hogan L.C. Designing for Performance. 1-е изд. Sebastopol, California : O'Reilly, 2015. 179 с. Текст : непосредственный.
- [5] SyncShare. Текст: электронный // SyncShare: [Электронный ресурс]. — URL: <https://syncshare.naloaty.me/> (дата обращения: 03.03.2025).
- [6] StatusNotifierItem. Текст: электронный // freedesktop.org : [Электронный ресурс]. — URL: <https://www.freedesktop.org/wiki/Specifications/StatusNotifierItem/> (дата обращения: 04.03.2025).
- [7] Cone, Matt The Markdown Guide / Matt Cone. 1-е изд, 2018. 85 с. Текст : непосредственный.
- [8] sync package - sync - Go Packages. Текст : электронный // The Go Programming Language: [Электронный ресурс]. — URL: <https://pkg.go.dev/sync> (дата обращения: 04.03.2025).
- [9] atomic package - sync/atomic - Go Packages. Текст: электронный // The Go Programming Language : [Электронный ресурс]. — URL: <https://pkg.go.dev/sync/atomic> (дата обращения: 04.03.2025).
- [10] Шелелугин И.М. Nakutest Современное и эффективное образовательное тестирование. Текст: электронный // Nakutest.org: [Электронный ресурс]. — URL: <https://hakutest.org/ru/> (дата обращения: 04.03.2025).
- [11] You Evan. VitePress: Vite & Vue Powered Static Site Generator / Evan You. Текст: электронный // VitePress: [Электронный ресурс]. — URL: <https://vitepress.dev/> (дата обращения: 04.03.2025).