

УДК 004.45

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ ОЦЕНКИ ПРОФЕССИОНАЛЬНЫХ НАВЫКОВ СОТРУДНИКОВ

Мурашко И.С., студент гр.281078

*Белорусский государственный университет информатики и радиоэлектроники,
Институт информационных технологий,
г. Минск, Республика Беларусь*

Скудняков Ю.А. – канд. техн. наук, доцент кафедры ИСиТ

Аннотация. В статье рассмотрены вопросы разработки веб-приложения для автоматизации процессов оценки профессиональных навыков сотрудников предприятия, представлен пользовательский интерфейс приложения, обоснован выбор средств разработки, перечислены недостатки аналогичных решений, представлены основные функциональные возможности программного средства. Серверная часть реализована на Java с использованием фреймворка Spring Boot, клиентская часть на TypeScript с использованием библиотеки React.

Ключевые слова. Веб-приложение, автоматизация тестирования, оценка персонала, Spring Boot, React, TypeScript.

Введение. В условиях цифровой трансформации экономики и стремительного развития технологий профессиональная подготовка персонала становится ключевым фактором повышения конкурентоспособности современных предприятий [1]. Современные организации сталкиваются с необходимостью постоянного мониторинга и оценки квалификации сотрудников, что определяется рядом критически важных аспектов. Во-первых, стремительная эволюция технологических стандартов и возникновение новых профессиональных компетенций требуют регулярного обновления методов оценки персонала. Во-вторых, традиционные подходы к аттестации, основанные на субъективных оценках и бумажном документообороте, демонстрируют свою ограниченную эффективность в условиях динамичной бизнес-среды [2].

Особое значение эти вопросы приобретают в IT-индустрии, где скорость устаревания знаний максимальна, а точность оценки профессиональных навыков напрямую влияет на успешность реализации проектов. Разработка автоматизированной системы тестирования позволяет решить указанные проблемы, обеспечивая стандартизированный, объективный и масштабируемый подход к оценке компетенций сотрудников.

Анализ существующих решений показал, что многие платформы для тестирования персонала недостаточно адаптированы под специфику различных предприятий. В ряде случаев функционал таких систем либо избыточен, либо не охватывает ключевые потребности организации. Это обуславливает необходимость разработки специализированного веб-приложения, способного эффективно решать задачи оценки профессиональных качеств сотрудников.

Целью данной работы является создание веб-приложения для автоматизации процессов оценки профессиональных навыков сотрудников предприятия. Проектирование системы базируется на принципах модульной архитектуры, реализованной на нескольких уровнях: на уровне бэкенда через четкое разделение на контроллеры, сервисы и репозитории; на уровне фронтенда – посредством компонентного подхода и выделения независимых модулей интерфейса [3,4].

В качестве технологического стека для серверной части выбран фреймворк Spring Boot, который предоставляет готовые решения для создания производственных приложений на Java, встроенные механизмы для разработки REST API, мощную систему инверсии контроля и зависимостей. Для клиентской части использованы библиотека React и язык TypeScript, что обеспечивает строгую типизацию данных и высокую поддерживаемость кода. Для управления состоянием применены библиотеки Zustand и TanStack React Query.

Архитектура системы построена по принципу «клиент-сервер» с использованием REST API для взаимодействия между компонентами. Серверная часть реализована на Spring Boot с использованием многослойной архитектуры (контроллеры, сервисы, репозитории). Клиентская часть представляет собой одностраничное приложение.

Практическая значимость результатов разработки заключается в возможности применения программного средства для автоматизации процессов оценки персонала на предприятиях различных отраслей.

Основная часть. Определим функциональные требования и назначение программного средства:

- управление пользователями и доступом с поддержкой ролевой модели (Администратор, Редактор, Пользователь);
- создание и редактирование тестовых заданий с поддержкой различных форматов вопросов (одиночный/множественный выбор, текстовый ответ, упорядочивание);
- категоризация вопросов по компетенциям и темам;

- настройка параметров тестирования (время выполнения, проходной балл, количество попыток);
- автоматическая и ручная проверка результатов тестирования;
- формирование аналитических отчетов по успеваемости сотрудников;
- просмотр истории тестирований и индивидуальной статистики.

Система проектировалась с учётом требований к расширяемости, безопасности и возможности одновременного прохождения тестирования для большого количества пользователей.

При запуске приложения пользователь проходит аутентификацию по персональному коду и паролю. В зависимости от назначенной роли пользователь получает доступ к соответствующему функционалу. Администраторы управляют пользователями и подразделениями, редакторы работают с тестовым контентом, обычные пользователи проходят тестирование и просматривают свои результаты.

На рисунке 1 представлен интерфейс главного окна после аутентификации, на котором доступен выбор вкладки.

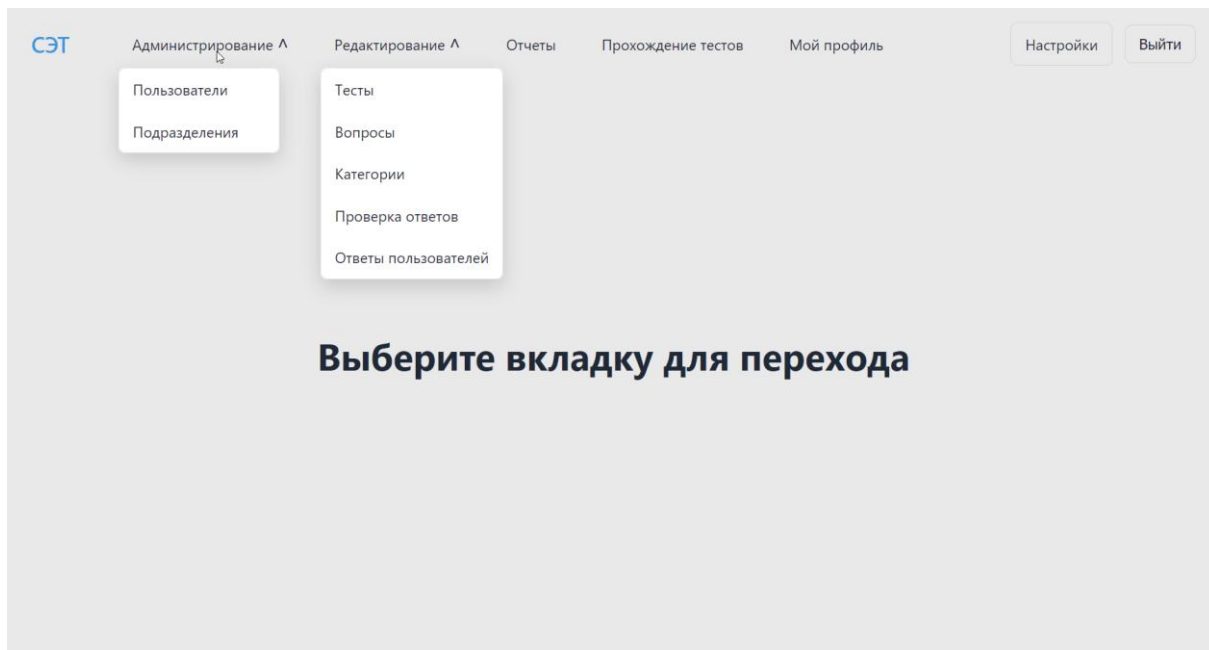


Рисунок 1 – Окно выбора рабочей вкладки

В верхней части окна расположена навигационная панель с доступными вкладками в зависимости от роли пользователя, а центральная часть отображает контент выбранной вкладки. Интерфейс реализован адаптивно и обеспечивает корректную работу на различных устройствах

Для создания тестов предусмотрены два подхода: фиксированный набор вопросов и динамический подбор вопросов из категорий. При фиксированном подходе редактор выбирает конкретные вопросы для теста. При динамическом подходе система случайно выбирает вопросы из указанных категорий в заданном количестве, что позволяет создавать различные варианты тестов для каждого пользователя.

Процесс тестирования включает следующие этапы:

- пользователь выбирает доступный тест из списка;
- система генерирует вопросы согласно стратегии теста;
- пользователь отвечает на вопросы с контролем времени;
- ответы автоматически сохраняются при переходе между вопросами;
- по завершении теста система вычисляет результат и предоставляет результат.

Методы проверки ответов различаются в зависимости от типа вопроса:

- для вопросов с выбором вариантов ответа проверка выполняется автоматически;
- для текстовых ответов предусмотрена ручная проверка редакторами;
- для заданий на упорядочивание применяется алгоритм сравнения последовательностей.

База данных спроектирована с учетом нормализации и включает 15 взаимосвязанных таблиц для хранения информации о пользователях, подразделениях, вопросах, тестах, категориях, результатах тестирования и ответах пользователей. Для обеспечения производительности разработаны индексы по ключевым полям таблиц.

Система безопасности включает аутентификацию на основе JWT-токенов, защиту от основных веб-уязвимостей и разграничение прав доступа в соответствии с ролевой моделью.

На рисунке 2 показан интерфейс создания вопроса с возможностью выбора типа вопроса, добавления вариантов ответа, прикрепления изображения и назначения категорий.

Рисунок 2 – Окно создания вопроса

Для улучшения пользовательского опыта были добавлены всплывающие подсказки, отображающиеся при наведении курсора на символ «(?)».

Сформируем модель расчета балла для ответа на вопрос. Для каждого вопроса определим:

$V = \{v_1, v_2, \dots, v_N\}$ – множество неверных вариантов;

$V_c = \{v \in V | v \text{ является верным}\}$ – множество верных вариантов;

$V_i = \{v \in V | v \text{ является неверным}\}$ – множество неверных вариантов;

При этом выполняются условия:

$V_c \cap V_i = \emptyset; V_c \cup V_i = V; |V_c| + |V_i| = N$.

При ответе пользователь выбирает подмножество $U \subseteq V$. Определим:

$C = |U \cap V_c|$ – количество выбранных верных вариантов;

$I = |U \cap V_i|$ – количество выбранных неверных вариантов;

Формула расчета сырого балла следующая:

$$S_{raw} = \frac{C-I}{N}.$$

Граничными значениями являются $S_{min} = -|V_i| / N$, при выборе пользователем всех неверных вариантов ответов и ни одного верного и $S_{max} = |V_c| / N$, при выборе пользователем всех верных вариантов ответов и ни одного неверного.

После вычисления S_{raw} производим нормирование к диапазону $[0; 1]$:

$$\begin{cases} 0, & S_{raw} \leq 0 \\ S_{raw}, & 0 < S_{raw} < 1 \\ 1, & S_{raw} \geq 1 \end{cases}$$

Итого модель оценивания:

- обладает полной нормированностью, поскольку итоговое значение балла в явном виде ограничивается интервалом $[0; 1]$;
- является монотонной по отношению к качеству ответа. Увеличение количества выбранных верных вариантов приводит к увеличению итогового балла, тогда как добавление неверных вариантов не может повысить оценку;
- с точки зрения симметричности симметрична на уровне сырого балла, поскольку вклад одного верного и одного неверного варианта равен по модулю и противоположен по знаку. Однако после применения нормирования симметрия нарушается вблизи границ шкалы вследствие насыщения значений на границах;
- полностью ограничена и устойчива к некорректным или случайным стратегиям прохождения теста, включая выбор всех вариантов ответа или отсутствие выбора;
- соблюдает свойство инвариантности к перестановке вариантов ответа, поскольку расчёт оценки основан исключительно на количестве выбранных верных и неверных вариантов и не зависит от их порядка или представления в интерфейсе;
- легко объяснима – каждый компонент формулы имеет однозначную семантику.

Итоговый результат тестирования определяется как сумма всех нормированных баллов за все вопросы теста. Тест считается успешно пройденным при достижении порогового значения, определяемого редактором при создании теста.

При проектировании системы автоматизированного тестирования ключевой задачей являлся выбор модели оценивания, обеспечивающей объективную, устойчивую и интерпретируемую оценку профессиональных знаний сотрудников. В качестве базового подхода была рассмотрена бинарная модель оценивания, при которой ответ признаётся либо полностью верным, либо неверным. Однако данный подход не позволяет учитывать частично корректные ответы и не обеспечивает достаточной дифференциации результатов, что особенно критично при оценке профессиональных компетенций

Использование предложенной модели, основанной на разности количества выбранных верных и неверных вариантов ответа, позволяет учитывать степень корректности ответа пользователя и снижать влияние случайного угадывания. Введение штрафа за выбор неверных вариантов стимулирует осознанный выбор ответов и повышает достоверность получаемых результатов по сравнению с моделями, в которых учитываются только верные варианты.

На рисунке 3 показан интерфейс тестирования с отображением текущего вопроса, вариантов ответа, таймера и индикатора прогресса.

Рисунок 3 – Вкладка «Попытка тестирования»

После завершения теста пользователь перенаправляется на вкладку обзора результатов. На данной вкладке показывается текущий результат.

Для каждого вопроса теста отображается блок, в котором перечисляются возможные варианты ответов, ответ данный пользователем и корректный ответ. В случае, если для вопроса добавлено

пояснение, то пользователь увидит его. Если пользователь не дал ответ на вопрос, или ответ не подлежит автоматическому оцениванию, то в статусе будет написано «Без ответа» или «На рассмотрении» соответственно. При получении балла в диапазоне (0,1), статус будет «Частично правильно». Ответы, получившие максимальный балл или получившие минимальный балл, будут иметь статус «Правильно» и «Неправильно» соответственно.

На рисунке 4 изображена вкладка просмотра результатов тестирования.

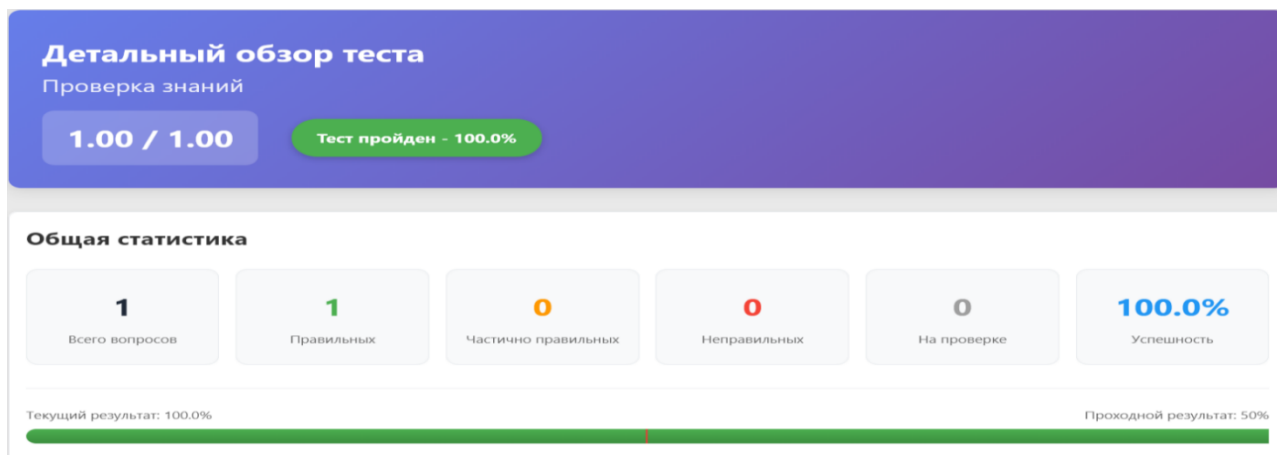


Рисунок 4 – Вкладка просмотра результатов

Заключение. Разработанное веб-приложение позволяет автоматизировать процессы планирования, проведения и анализа результатов тестирования профессиональных навыков сотрудников предприятия. Система обеспечивает стандартизированный, объективный и масштабируемый подход к оценке компетенций персонала, устраняя зависимость от бумажного документооборота и субъективных оценок. Ключевым достоинством используемой архитектуры является наличие веб-интерфейса, с которыми могут работать браузеры на смартфонах и компьютерах, не требуя установки дополнительного программного обеспечения.

Список использованных источников:

1. Прокопьев, А. В. Обучение персонала как фактор организационного развития / А. В. Прокопьев, Т. Н. Чернышова // *Cifra. Экономика*. – 2023. – №3. – Режим доступа: <https://doi.org/10.23670/ECNMS.2023.3.4>.
2. Шевнина, Ю.С. Автоматизация оценки профессиональных качеств и компетенций сотрудников предприятия / Ю.С. Шевнина, В.В. Константинов // *Вестник Астраханского гос. техн. ун-та*, 2021, №2(72), С. 7-14. DOI:10.24143/1812 – 9498-2021-2-7-14.
3. Мартин, Р. Чистая архитектура. Искусство разработки программного обеспечения / Р. Мартин. – Санкт-Петербург. : Питер, 2021. – 384 с.
4. Блох, Д. Java. Эффективное программирование / Д. Блох – Санкт-Петербург: Диалектика, 2019. – 464 с.

UDC 004.45

WEB APPLICATION FOR AUTOMATION OF EMPLOYEE PROFESSIONAL SKILLS ASSESSMENT

Murashka I.S.

*Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus*

Skudnyakov Yu.A. – Candidate of Engineering Sciences, Associate Professor

Annotation. This article discusses the development of a web application for automating employee professional skills assessment processes. It presents the application's user interface, justifies the choice of development tools, lists the shortcomings of similar solutions, and outlines the software's key functionality. The server-side component is implemented in Java using the Spring Boot framework, while the client-side component is written in TypeScript using the React library.

Keywords. Web application, test automation, personnel assessment, Spring Boot, React, TypeScript.