

ждому узлу соответствует отдельная директория, содержащая все необходимые библиотеки, конфигурационные файлы и исполняемый модуль.

Для разрешения зависимостей Python-обработчиков требуется скачать необходимые папки с зависимостями, включая клонирование репозитория Whisper и SileroVAD с GitHub. При скачивании зависимостей так же необходимо скачать torch 2.1 и torchaudio 2.1, так как это позволяет разрешить проблемы зависимостей бэкенда обработки аудио у SileroVAD. При запуске процессов обработчиков воркер обязан установить переменные среды на те, что находятся в папке зависимостей с помощью свойства `ProcessStartInfo.EnvironmentVariables`.

Таким образом, предложенная система демонстрирует эффективный подход к организации распределенной обработки аудиоданных в среде Windows. Она сочетает в себе простоту развертывания, гибкость взаимодействия компонентов и высокую отказоустойчивость, что делает ее перспективным решением для задач транскрибации больших аудиофайлов.

#### Литература

1. Silero Team. Silero VAD: pre-trained enterprise-grade Voice Activity Detector. – URL: <https://github.com/snakers4/silero-vad>.
2. Radford A. et al. Robust Speech Recognition via Large-Scale Weak Supervision. – URL: <https://github.com/openai/whisper>.
3. Microsoft. System.Collections.Concurrent Namespace. – URL: <https://learn.microsoft.com/en-us/dotnet/api/system.collections.concurrent>.

## ИНТЕРАКТИВНЫЙ ПРОГРАММНЫЙ ТРЕНАЖЕР ПО ТЕМЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ С ИСПОЛЬЗОВАНИЕМ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

А. А. Готченя

Учреждение образования «Гомельский государственный технический  
университет имени П. О. Сухого», Республика Беларусь

Научный руководитель О. Д. Асенчик

*Разработан интерактивный программный тренажер по информационной безопасности на основе больших языковых моделей (LLM). Система выступает в роли виртуального экзаменатора: генерирует вопросы из загруженной базы нормативных документов, анализирует развернутые ответы обучаемых на естественном языке, при неверном ответе формирует наводящий вопрос – не раскрывая правильного решения – и по завершении сессии выставляет итоговую оценку с развернутым анализом. Применение технологии дополненной генерации (RAG) исключает галлюцинации и привязывает обучение к актуальной внутренней документации организации.*

**Ключевые слова:** большие языковые модели, информационная безопасность, дополненная генерация (RAG), интерактивный тренажер.

Информационная безопасность (ИБ) является одной из наиболее динамично развивающихся областей современных информационных технологий. Рост числа кибератак, утечек данных и инцидентов, связанных с действиями сотрудников, формирует устойчивый запрос на системы обучения, способные не просто информировать пользователей, но и формировать у них устойчивые практические навыки.

Традиционные методы подготовки – статичные тесты с выбором правильного варианта – демонстрируют принципиальное ограничение: пользователи запоминают

правильные варианты ответов механически, но теряются при столкновении с нетипичными сценариями фишинговых атак, утечек данных или инцидентов социальной инженерии. Появление больших языковых моделей открывает принципиально новые возможности для автоматизации обучения: в отличие от статических тестовых систем, *LLM* способны вести диалог на естественном языке, анализировать развернутые ответы и формулировать наводящие вопросы вместо немедленного раскрытия правильного ответа.

Таким образом, создание интерактивного тренажера, реализующего диалоговое обучение с ролью виртуального тьютора и привязкой к корпоративной базе знаний, является актуальной задачей как для образовательных учреждений, так и для организаций, заинтересованных в повышении осведомленности персонала в области ИБ.

На современном рынке существует ряд решений для обучения в области информационной безопасности. *KnowBe4* – одна из наиболее широко используемых платформ – предлагает обширную библиотеку видеоуроков и симуляторов фишинговых атак, однако строится на пассивном формате: пользователь просматривает контент и проходит тест с выбором ответа. Система не ведет диалога с обучаемым и не предоставляет персонализированную обратную связь. Кроме того, *KnowBe4* является закрытым *SaaS*-сервисом, что исключает интеграцию собственной базы знаний организации.

Образовательные платформы общего назначения – *Coursera*, *Stepik*, *Cybrary* – предоставляют структурированные курсы по ИБ, но ориентированы на широкую аудиторию и не предполагают адаптации контента под специфику конкретной организации. Механизм проверки знаний сводится к тестам с вариантами ответов; интерактивный диалог в формате «вопрос – свободный ответ – наводящий вопрос» отсутствует.

Платформы *TryHackMe* и *Hack The Box* предоставляют реалистичные виртуальные лаборатории для отработки технических навыков, однако ориентированы на подготовку специалистов-практиков в области кибербезопасности, а не на массовое повышение осведомленности сотрудников. Они не поддерживают загрузку корпоративных политик в качестве базы знаний и не имеют модуля управления диалогом с ролью тьютора.

Универсальные – языковые модели – *ChatGPT*, *Claude*, *Gemini* – теоретически способны вести диалог в формате обучения, однако обладают рядом принципиальных ограничений: не имеют доступа к внутренней базе знаний организации, не реализуют заданный педагогический сценарий с управлением попытками и начислением баллов, а также не ограничивают пользователя от получения прямых ответов – обучаемый может явно попросить модель раскрыть решение, что нивелирует педагогический эффект.

Таким образом, разрабатываемый тренажер занимает нишу, не закрытую ни одним из существующих на рынке решений: система сочетает диалоговое обучение на основе *LLM*, привязку к корпоративной базе знаний через *RAG* [1], гибкую настройку педагогических сценариев и возможность локального развертывания.

Целью работы является разработка интерактивного программного тренажера по информационной безопасности с использованием больших языковых моделей. Тренажер выступает в роли виртуального экзаменатора, который генерирует вопросы на основе загруженной базы нормативных документов и учебных материалов, оценивает свободные ответы обучаемых, при необходимости задает наводящие вопросы и по завершении сессии выставляет итоговую оценку с развернутым анализом.

Практическая значимость работы определяется возможностью применения сис-

темы для обучения сотрудников организаций основам ИБ с опорой на внутреннюю нормативную документацию. Система поддерживает как облачный доступ через *OpenRouter*, так и локальное развертывание через *Ollama*, что обеспечивает применимость в организациях с ограничениями на передачу данных во внешние сервисы.

Система построена по двухзвенной клиент-серверной архитектуре. Серверная часть реализована на платформе *.NET 10 (ASP.NET Core Web API)* [2] и содержит всю бизнес-логику: управление экзаменационными сессиями, взаимодействие с языковыми моделями, индексирование базы знаний и хранение данных. Клиентская часть представляет собой одностраничное браузерное приложение на *Blazor WebAssembly* [3], ответственное за отображение диалогового интерфейса.

Логика проведения тренинга реализована в виде сериализуемого конечного автомата на базе *Semantic Kernel Process Framework* [4]. Процесс состоит из восьми шагов: выбор темы, генерация вопросов, постановка вопроса, фильтрация сообщений от пользователя, оценивание ответа, генерация подсказки, итоговое оценивание и отображение сообщений. Принципиально важно, что переходы между шагами управляются серверным кодом, а не языковой моделью – *LLM* отвечает исключительно за генерацию текста. После каждого хода полное состояние процесса сериализуется в *JSON* и сохраняется в базе данных, что позволяет обслуживать произвольное число одновременных сессий без хранения данных в памяти сервера.

Вся генерация текста выполняется через четыре специализированных *AI*-агента на базе *Microsoft Semantic Kernel* [5]: агент генерации вопросов, агент оценивания ответов, агент генерации подсказок и агент итогового оценивания. Каждый агент описан в отдельном *YAML*-файле с промптом, что позволяет изменять инструкцию агента без правки кода.

Для хранения базы знаний используется векторное хранилище *Qdrant*. Пайплайн индексирования включает четыре этапа: извлечение текста из файлов форматов *PDF*, *RTF*, *DOCX*, очистка от артефактов форматирования, семантическая разбивка на фрагменты (чанки по 500 токенов с перекрытием 100 токенов) и батчевая векторизация с помощью эмбединговой модели. При этом структура элемента в векторной базе данных зависит от типа документа: если фрагмент принадлежит статье нормативно-правового акта, элемент хранит как сам чанк размерностью 500 токенов, так и полный текст соответствующей статьи – именно он передается в качестве контекста при генерации вопросов и оценивании ответов; если же документ не является нормативным актом, элемент содержит только чанк без ссылки на родительский фрагмент. Для хранения структурированных данных – учетных записей, истории сессий, журналов диалогов и оценок – применяется реляционная СУБД *SQLite* через *Entity Framework Core*.

Безопасность системы обеспечивается на двух уровнях. Для минимизации галлюцинаций применяется системный промпт, запрещающий *LLM* использовать знания вне предоставленного *RAG*-контекста. Для защиты от атак типа *Prompt Injection*, когда пользователь пытается перенастроить роль тьютора и вынудить систему раскрыть правильный ответ, реализована фильтрация входящих сообщений.

В ходе работы разработан и реализован полный программный комплекс интерактивного тренажера. Проведенный сравнительный анализ подтвердил, что существующие решения – платформы *KnowBe4*, *Coursera*, *TryHackMe* и универсальные *LLM*-чат-боты – не обеспечивают совместно диалогового обучения, привязки к корпоративной базе знаний и гибкой настройки педагогических сценариев. Разработанная система устраняет каждый из этих недостатков.

Реализованы все ключевые подсистемы: пайплайн индексирования документов, модуль семантического поиска, оркестратор экзаменационного процесса на базе ко-

нечного автомата, четыре специализированных *AI*-агента и чат-интерфейс на *Blazor WebAssembly*. Система успешно протестирована на сценариях обучения по темам информационной безопасности с использованием реальных нормативных документов.

Разработанный интерактивный тренажер реализует диалоговый формат обучения, принципиально отличающийся от традиционных тестовых систем: при неверном ответе система не раскрывает правильного решения, а задает наводящий вопрос, побуждая обучаемого к самостоятельному нахождению ответа. Применение технологии *RAG* обеспечивает привязку всех генерируемых вопросов и оценок к актуальному содержимому загруженных документов. Реализация экзаменационного процесса в виде сериализуемого конечного автомата гарантирует предсказуемость педагогического сценария и масштабируемость системы. Гибкость настройки сценариев и поддержка локального развертывания делают тренажер применимым как в учебных заведениях, так и в организациях с ограничениями на передачу данных во внешние облачные сервисы.

#### Литература

1. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks / P. Lewis, E. Perez, A. Piktus [et al.] // *Advances in Neural Information Processing Systems*. – 2020. – Vol. 33. – P. 9459–9474.
2. Microsoft. ASP.NET Core documentation. – URL: <https://learn.microsoft.com/en-us/aspnet/core/> (дата обращения: 15.03.2026).
3. Microsoft. Blazor WebAssembly documentation. – URL: <https://learn.microsoft.com/en-us/aspnet/core/blazor/> (дата обращения: 15.03.2026).
4. Microsoft. SK Process Framework Overview. – URL: <https://learn.microsoft.com/en-us/semantic-kernel/frameworks/process/process-framework> (дата обращения: 20.03.2026).
5. Microsoft. Semantic Kernel Documentation. – URL: <https://learn.microsoft.com/en-us/semantic-kernel/> (дата обращения: 20.03.2026).

### ИНТЕРАКТИВНАЯ СИСТЕМА ПЛАНИРОВАНИЯ И УЧЕТА ИНДИВИДУАЛЬНОЙ УЧЕБНОЙ НАГРУЗКИ ПРЕПОДАВАТЕЛЯ НА БАЗЕ ПЛАТФОРМЫ .NET 10

И. Ю. Тимошенко

Учреждение образования «Гомельский государственный технический  
университет имени П. О. Сухого», Республика Беларусь

Научный руководитель О. Д. Асенчик

*Описана интерактивная система планирования и учета индивидуальной учебной нагрузки преподавателя, предназначенная для замены электронных таблиц и снижения трудоемкости отчетности. Реализовано веб-приложение на платформе .NET 10 (Blazor Server) с гибридной моделью хранения данных (включая JSON) и механизмом «снимков» для сохранения архивной достоверности утвержденных планов. Представлены модуль импорта нагрузки из XLSX, алгоритм месячного распределения часов, учет фактического проведения и генерация отчетности; работоспособность решения подтверждена модульными и интеграционными тестами. Приложение подходит как для персонального использования, так и для внедрения в масштабе университета.*

**Ключевые слова:** *Blazor Server, .NET 10, учебная нагрузка, информационная система, автоматизация документооборота, Entity Framework Core.*

В условиях современной цифровой трансформации образовательной среды задачи оперативного управления и контроля деятельности профессорско-препода-