

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА ЕДИНОЙ МЕЖВУЗОВСКОЙ АКАДЕМИЧЕСКОЙ
ПЛАТФОРМЫ НА ОСНОВЕ СТАНДАРТА СМІ5**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Прыткова Дмитрия Александровича

Научный руководитель
доцент, к. ф.-м. н.

М. И. Сафрончик

Заведующий кафедрой
доцент, к. ф.-м. н.

С. В. Миронов

Саратов 2025

ВВЕДЕНИЕ

Актуальность темы. Цифровизация становится неотъемлемой частью современного образования, обеспечивая персонализацию обучения, гибкость образовательных траекторий и эффективный сбор данных о достижениях обучающихся. Одним из ключевых инструментов для реализации этой задачи является стандарт СМІ5, позволяющий детализировано анализировать учебную активность и управлять курсами. Внедрение СМІ5 в масштабах национальной образовательной системы может стать основой для создания всероссийской академической платформы, обеспечивающей сквозной мониторинг и поддержку образовательного процесса.

Актуальность работы обусловлена необходимостью интеграции разрозненных образовательных систем, повышения качества сбора и анализа данных об обучении, а также создания единого пространства для академической мобильности. СМІ5, в отличие от традиционных стандартов, предоставляет более гибкие возможности для фиксации учебного опыта, что делает его перспективным решением для масштабируемых образовательных платформ.

Цель бакалаврской работы — разработка единой академической платформы, которая предоставляет университетам возможность размещать учебные материалы, обеспечивает сбор, хранение и анализ данных об учебной деятельности.

Поставленная цель определила **следующие задачи:**

1. проанализировать стандарты электронного образования для создания академических платформ;
2. исследовать существующие решения в области цифрового образования;
3. разработать образовательную платформу, позволяющую различным университетам размещать обучающие курсы в едином пространстве и поддерживать полный цикл работы с ними, включая анализ данных об учебной деятельности.

Структура и объем работы. Бакалаврская работа состоит из введения, трех разделов, заключения, списка использованных источников и трех приложений. Общий объем работы — 50 страниц, из них 43 страницы — основное содержание, включая 26 рисунков, цифровой носитель в качестве приложения, список использованных источников информации — 20 наименований.

1 Обзор современного цифрового образования

1.1 Стандарты электронного образования

SCORM.

SCORM (Sharable Content Object Reference Model) долгие годы оставался основным стандартом для электронных курсов. Его ключевое преимущество — строгая стандартизация формата упаковки контента в ZIP-архив, что обеспечивает совместимость между разными платформами. Однако SCORM разрабатывался в эпоху, когда основным носителем учебного контента были CD-диски, что наложило существенные ограничения на его архитектуру.

Главные технические ограничения SCORM проявляются в трех аспектах: работа только в браузерной среде, отсутствие поддержки режима без доступа к интернету и невозможность фиксации сложных учебных активностей, что делает SCORM непригодным для современных адаптивных систем обучения.

Experience API.

Ключевое нововведение xAPI — концепция Learning Record Store (LRS) — специализированного хранилища учебных данных, независимого от самой платформы. Это позволяет создавать сложные аналитические системы, агрегирующие данные из множества источников. Формат данных xAPI, называемые «statements», основан на трех сущностях «Актор», «Действие» и «Объект» и поддерживает произвольные метаданные.

Однако гибкость xAPI стала и его слабым местом. Отсутствие стандартных словарей глаголов и типов активностей приводит к проблемам совместимости между разными системами. Кроме того, xAPI не предлагает готовых решений для структурирования учебных программ, что осложняет его использование в академической среде с жесткими учебными планами.

CMi5.

CMi5 появился как ответ на ограничения xAPI в корпоративном и академическом обучении. Технически это надстройка над xAPI, добавляющая недостающую структуру через несколько ключевых компонентов: Assignable Units (AU) как минимальные единицы контента, механизм запуска через платформы и обязательные поля данных. При этом сохраняются все преимущества xAPI в части гибкости и расширяемости.

Среди рассмотренных стандартов электронного обучения CMi5 представляет собой наиболее сбалансированное решение для академических платформ.

В отличие от SCORM, он поддерживает современные форматы обучения (десктопные или мобильные приложения, офлайн-активности), а по сравнению с xAPI предлагает четкую структуру курсов и стандартизированную отчетность.

1.2 Ключевые требования к платформе на базе CMi5

Архитектурные требования

При проектировании архитектуры платформы первостепенное значение имеет выбор и настройка системы управления обучением (LMS). Современная LMS должна не только обеспечивать базовые функции управления курсами, но и обладать расширенными возможностями интеграции.

Хранилище записей обучения (LRS) выступает центральным узлом для сбора и анализа данных. В отличие от традиционных решений, LRS для CMi5 должно обрабатывать значительно большие объемы структурированной информации, сохраняя при этом высокую производительность.

Сервер курсов представляет собой специализированное решение для хранения и распространения учебного контента. Его ключевая особенность — поддержка динамической генерации структур для каждого учебного курса. Важным аспектом также является обеспечение безопасного доступа к образовательному контенту.

Требования к данным

CMi5 использует расширенную модель данных xAPI, где каждая учебная активность описывается в виде структурированного statement. В отличие от базового xAPI, CMi5 накладывает строгие требования на формат этих записей, что особенно важно для обеспечения согласованности данных в масштабах национальной образовательной системы. Каждый statement должен содержать обязательные поля, описывающие участника, выполненное действие, объект взаимодействия и полученный результат.

Функциональные требования

Для преподавательского состава платформа должна предоставлять комплексную среду разработки курсов. Современный редактор должен поддерживать создание интерактивных материалов с возможностью встраивания мультимедийного контента и интерактивных элементов.

Студенческий интерфейс необходимо проектировать с учетом принципов современного дизайна для образовательных платформ. Помимо стандартного функционала прохождения курсов, важно предусмотреть систему поиска инте-

ресующих курсов. Все элементы интерфейса должны быть доступны с различных устройств, обеспечивая непрерывность учебного процесса.

Административный модуль требует реализации системы ролей и прав доступа, учитывающей иерархическую структуру образовательных учреждений. Все административные функции должны быть защищены от неправомерного доступа.

1.3 Существующие решения в области цифрового образования

Moodle

Moodle является одной из самых распространенных систем дистанционного обучения. Платформа предлагает широкий набор инструментов для создания онлайн-курсов, управления учебным процессом и организации взаимодействия между преподавателями и обучающимися.

Однако, при всех своих достоинствах Moodle обладает рядом существенных недостатков, которые ограничивают её применение в качестве основы для единой межвузовской академической платформы. Среди них можно выделить устаревший пользовательский интерфейс, архитектура платформы, не обеспечивающая необходимой производительности при масштабировании, сильная зависимость от сторонних плагинов, зависимость от зарубежных репозиторий и международного сообщества разработчиков.

Портал «Современная цифровая образовательная среда»

Представляет собой агрегатор онлайн-курсов от ведущих российских вузов и образовательных платформ. Система выполняет роль единого окна доступа к цифровым образовательным ресурсам. Главный архитектурный недостаток СЦОС — отсутствие единой платформы для создания и прохождения курсов. Технически портал реализован как платформа, которая не содержит собственных курсов, а перенаправляет пользователей на внешние ресурсы, что добавляет еще одну проблему — несовместимость технологических решений, так как разные платформы используют различные стандарты.

В сравнении с текущей реализацией СЦОС, предлагаемая платформа на базе СМІ5 может обеспечить единое образовательное пространство, централизованное хранение всех учебных материалов, инструменты разработки контента и полный цикл работы с курсами. Такой подход сохраняет преимущества агрегирования образовательного контента и устраняет ключевые недостатки текущей реализации СЦОС.

2 Архитектура приложения

Основополагающим требованием к работе академической платформы является наличие следующих компонент:

1. **Learning Management System (LMS)** — система управления обучением. Ее основные функции заключаются в размещении учебных материалов, управлении пользователями, оценки работ, аналитике, отчетности.
2. **Assignable Units (AU)** — минимальная единица обучающего контента, которую можно как-то оценить. Представляет собой отдельные учебные модули, например, курсы, тесты или задания.
3. **Learning Record Store (LRS)** — принимает и хранит данные, делает их доступными для LMS.

Для реализации этих компонент хорошо подходит трехуровневая архитектура — популярная модель проектирования программного обеспечения, которая разделяет приложение на три логических и физических уровня:

1. **Презентационный уровень** — отвечает за взаимодействие с пользователем, включает в себя интерфейсы веб-страниц, мобильных или десктопных приложений.
2. **Бизнес-логика** — обеспечивает связь между клиентом и базой данных, обрабатывая запросы от пользователей.
3. **Уровень данных** — хранит данные.

Такая архитектура одновременно удобна в разработке и хорошо масштабируема, так как изменение одного уровня не требует переделки других и каждый уровень можно масштабировать отдельно. Также она имеет хорошую безопасность, поскольку клиент не имеет прямого доступа к данным.

Таким образом, LMS вместе с AU можно реализовать на презентационном уровне в качестве клиентского веб или десктоп приложения, а LRS на уровне бизнес-логики и уровне данных в качестве серверного приложения с привязкой к базе данных.

Технологии для разработки клиентского приложения — Vite, React, Redux Toolkit, Tailwind CSS, Heroicons, Axios, NW.js.

Технологии для разработки серверного приложения — Nest.js, MongoDB, Mongoose, Class-validator, JSON Web Tokens.

3 Разработка академической платформы

Страницы регистрации и авторизации.

Чтобы получить доступ к образовательным материалам и персонализированному обучению необходимо дать пользователям возможность регистрации и входа на платформу. Для достижения этой задачи был разработан класс пользователей со следующими полями: фамилия, имя, отчество, уникальный никнейм, пароль, электронная почта, роль (студент, преподаватель или администратор), массив курсов, принадлежность к университету и факультету, а также путь к изображению для аватара.

В рамках регистрации на платформу пользователю предлагается ввести такие поля, как фамилия, имя, отчество, уникальный никнейм, пароль и электронная почта. Остальные будут выставлены в базовое состояние на сервере, а пароль хэширован для надежного хранения. Каждое входное поле проверяется с помощью библиотеки class-validator.

Для входа на платформу достаточно ввести свой уникальный никнейм или почту и пароль. После входа пользователю выдается токен доступа, благодаря которому он может оставаться в системе определенное время без повторного входа.

Страница университетов.

После входа в систему пользователю предлагается список университетов, которые привязаны к платформе. Ему видна такая информация, как название университета, город нахождения, описание и изображение.

Был разработан класс университетов со следующими полями: полное название, уникальное короткое название (для использования в ссылках), местонахождение, описание и список факультетов, которые будут продемонстрированы после перехода на определенный университет.

Добавление университета предполагается соответствующим запросом к разработчикам или владельцам платформы.

Страница факультетов.

После перехода на страницу отдельного университета пользователь видит более подробную информацию о нем, такую как название, описание, год основания, расположение, количество студентов, достижения, а также список факультетов.

Был разработан класс факультетов со следующими полями: полное на-

звание, уникальное короткое название (для использования в ссылках), описание, количество курсов, глава факультета, контактная информация, достижения, список курсов факультета.

Страница администратора.

Для управления информацией в определенном университете требуется администратор, который назначается самим университетом запросом к разработчикам или владельцам академической платформы.

Чтобы исполнять обязанности администратора была создана страница «Панель администратора», на которой пользователь, имея соответствующую роль, может управлять данными своего университета.

Панель разделена на две части. Первая — управление факультетами. При открытии панели администратор видит список факультетов. В нем он может открыть редактор и изменить информацию о факультете или удалить его. Также в этой части присутствует кнопка «Добавить факультет», нажатие на которую откроет окно редактора факультетов, где администратор может добавить новое подразделение.

Во второй части панели администратор может производить управление пользователями. Он видит всех пользователей своего университета, и может выдавать им роли студента, преподавателя или администратора, а также назначать факультет, что необходимо для преподавателей для создания курсов на своем факультете. Интерфейс спроектирован так, что можно найти пользователя по имени или почте, отсортировать их по ролям или факультетам, что делает процесс управления более гибким и быстрым.

Страницы курсов.

При переходе на определенный факультет, пользователь может видеть страницу с информацией о нем, а также о размещенных курсах.

Для определения курсов была разработана специальная схема базы данных, включающая такие поля, как короткие уникальные имена университета, факультета и самого курса, название курса, пароль доступа, описание, преподаватели, теги, список групп студентов, список студентов, массив модулей курса.

На данной странице работает поиск курсов, который может искать как по их названию, так и по описанию, тегам и преподавателям. Помимо этого, для преподавателей и администраторов доступна возможность создания курсов по определенному шаблону и загрузка курса в виде JSON файла, что позволяет

мгновенно перенести его из другой платформы.

Также для более удобного поиска различных курсов со всей страны была добавлена страница, объединяющая их в один список. Таким образом, человек может оперативно найти интересующий его курс от любого университета.

Страница отдельного курса.

Возможность проходить курсы является одной из ключевых для академической платформы. Для доступа к курсу необходимо получить пароль от преподавателя, при вводе которого пользователь автоматически записывается на курс и имеет право на его прохождение. При этом он автоматически определяется в группу «Н.О.» (не определен).

На странице курса можно наблюдать всю основную информацию о нем. Здесь же расположены модули — особые единицы, делящие курс на разделы. Каждый модуль содержит заголовок, текстовую часть или описание, задания, тесты и прикрепленные файлы.

Страница для преподавателя и для студента немного отличается. Преподаватели могут различно изменять курс, добавлять новые модули, задания, тесты, прикреплять файлы. Одной из особенностей платформы является экспорт курса в JSON файл, что позволяет сохранить его локально на компьютере, или использовать на других платформах. Отсюда же можно перейти к странице с информацией о студентах, дающей полную сводку о прохождении курса.

От лица студента на странице курса можно ознакомиться с разделами или прикрепленными материалами и перейти к выполнению соответствующих заданий или тестов.

Страница заданий.

Страница заданий дает возможность выкладывать в определенный модуль курса задания.

Был создан отдельный класс, содержащий такие поля, как заголовок, описание, максимальная оценка, срок сдачи, прикрепленные файлы и уникальное короткое название, которое создается автоматически для использования в ссылках.

Страница заданий, аналогично остальным элементам курса, имеет два вида: вид от лица студента и от лица преподавателя.

Находясь на странице задания, студент видит всю необходимую информацию о нем, может скачать прикрепленные материалы, а также оставить свое

решение. После отправки решения, если оно еще не было оценено, студент может его изменить, но при этом срок сдачи также обновляется.

Преподаватель, находясь на странице, может менять различные поля задания, такие как срок сдачи или описание, а также прикреплять дополнительные файлы. Также преподаватель видит отправленные работы на проверку от студентов и может их оценить.

Страница тестирования.

Страница тестирования дает возможность выкладывать в определенный модуль курса тесты.

Был создан отдельный класс, содержащий такие поля, как заголовок, описание, максимальная оценка, срок сдачи, список вопросов и уникальное короткое название, которое создается автоматически для использования в ссылках.

Страница тестирования, аналогично странице заданий, имеет два вида: вид от лица студента и от лица преподавателя.

Преподаватель может разными способами изменять тесты, добавлять, обновлять или удалять вопросы, настраивать количество возможных и правильных ответов, выставять для каждого вопроса свой ценностный балл.

Студент может проходить тест, если его время сдачи еще не закончилось. После прохождения теста студент видит свой результат, и больше не может пройти его заново.

Страница отслеживания студенческой активности.

У преподавателя есть возможность отслеживать результаты студентов по мере прохождения курса.

Был создан отдельный класс студентов на курсе, содержащий такие поля, как полное имя, никнейм, итоговое количество баллов, группа, список сданных работ.

Преподаватель может создавать, настраивать различные группы и определять в них студентов. На странице отображается таблица студентов, в которой отображаются имена студентов, их группа, баллы за выполненные задания и итоговые баллы. Таблицу можно сортировать как по группам или именам студентов, так и по количеству баллов.

Также на странице есть возможность экспортировать данные в CSV файл для дальнейшего использования в агрегаторах таблиц, например Microsoft Excel, чтобы преподаватель мог оставлять свои пометки и отчеты по студентам.

Страница профиля пользователя.

Каждый пользователь имеет возможность отслеживать свои результаты на странице личного профиля. Тут он видит вкладки с курсами, на которые записан, и результаты своей работы в удобной табличной форме.

Десктопная версия.

Версия платформы для рабочего стола может улучшить пользовательский опыт путем работы в режиме без интернета, сохраняя курсы локально на компьютер.

Для реализации сборки такой версии используется фреймворк NW.js. Он собирает приложение для операционной системы Windows и генерирует исполняемый файл .exe в 64 разрядной системе.

Чтобы адаптировать приложение под веб и десктопную версии можно использовать глобальную переменную «window». Если приложение запущено на рабочем столе, то в объекте «window» будет находиться сущность «nw», которая и определяет версию. Так, например, благодаря этой информации, можно определить тип маршрутизации платформы. Веб версия должна использовать функцию createBrowserRouter, тогда как десктопная — createHashRouter.

Таким образом, на странице каждого курса в десктопной версии располагается кнопка «Сохранить локально», что позволяет сохранить текущую информацию о курсе на рабочий стол для дальнейшего использования.

Если сервер не отвечает или есть проблемы с интернет соединением, то у пользователя есть возможность просмотреть скачанные ранее материалы курса офлайн, благодаря специально разделу на боковой панели.

ЗАКЛЮЧЕНИЕ

В ходе выпускной квалификационной работы была проведена комплексная разработка академической платформы на базе стандарта СМІ5, включающая анализ современных технологических решений и реализацию ключевых функциональных компонентов. Основное внимание уделялось созданию архитектуры, обеспечивающей надежное хранение учебных данных и удобный интерфейс для всех категорий пользователей.

Особое значение имела проработка механизмов аутентификации и авторизации, обеспечивающих безопасный доступ к образовательным ресурсам. Использование современных технологий позволило реализовать надежную систему управления пользователями с поддержкой различных ролей и прав доступа. Разработанное решение успешно решает проблему фрагментации данных, характерную для существующих платформ-агрегаторов.

Ключевым преимуществом предложенной платформы является ее универсальность: она объединяет образовательные ресурсы вузов в единое пространство, позволяя студентам и преподавателям со всей страны получать доступ к официальным курсам ведущих университетов, что открывает новые возможности для академической мобильности, непрерывного обучения и стандартизации образовательных процессов на национальном уровне.

Перспективы развития платформы включают дальнейшее расширение функциональности за счет интеграции с внешними образовательными системами и внедрения инструментов аналитики данных. Реализованная платформа создает прочную основу для построения единого образовательного пространства, соответствующего современным требованиям электронного обучения.