

Клиентская часть программного комплекса и интерфейс приложения реализуется с помощью технологии *Blazor Webassembly*. Выбранная технология является частью экосистемы *.Net Core* и позволяет разрабатывать быстрый и удобный интерфейс клиента. Данный фреймворк обладает полной совместимостью с решениями *ASP.Net Core*. Клиентская часть приложения реализуется на языке программирования *C#*, который подходит для разработки клиентских приложений и имеет все необходимые инструменты для реализации.

Применение разработанного программного комплекса позволит автоматизировать процесс мониторинга исполнения техники безопасности в опасных зонах на промышленном предприятии, а также улучшить качество контроля на производстве. Внедренная система позволит предотвратить воздействия на работающих опасных производственных факторов, являющихся причиной травм или внезапного резкого ухудшения здоровья.

WEB-ПРИЛОЖЕНИЕ ПО УЧЕТУ РЕЗУЛЬТАТОВ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ СТУДЕНТОВ ГГТУ ИМ. П. О. СУХОГО

Д. В. Староверова

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

Научный руководитель Н. В. Ермалинская

Определены необходимые функциональные требования, рассмотрены технологии реализации и архитектура проектируемого web-приложения по учету результатов учебной деятельности студентов ГГТУ им. П. О. Сухого.

Ключевые слова: *web-приложение, учет учебной деятельности студентов.*

Введение. В современном образовательном процессе внедрение цифровых технологий играет ключевую роль, обеспечивая эффективное управление и анализ учебной деятельности студентов. Предлагаемое *web-приложение* решает ряд существенных проблем, с которыми сталкиваются преподаватели и администрация учебных заведений. Одной из основных проблем является необходимость оперативного и точного учета успеваемости студентов. Кроме того, такое приложение позволит автоматизировать процесс формирования отчетов и аналитики по результатам обучения, что значительно повысит эффективность работы преподавателей и специалистов, сопровождающих учебный процесс.

Таким образом, целью проводимых исследований является разработка *web-приложения* по учету результатов учебной деятельности студентов ГГТУ им. П. О. Сухого.

Основная часть. Учет учебной деятельности студентов включает в себя ряд задач: работа с приказами на зачисление(отчисление) студентов, подготовка распоряжений о переводе на следующий курс, ведение личных и учебных карт студентов, учет и анализ успеваемости, составление протоколов по результатам защиты дипломных работ в государственной экзаменационной комиссии, подготовка документов об образовании.

С учетом вышесказанного, в рамках проектируемого *web-приложения* будут автоматизированы следующие процессы:

- учет студентов, зачисленных на факультет;
- учет результатов промежуточной и итоговой аттестации студентов;

– ведение документации о результатах учебной деятельности студентов (учебная карточка, зачетная книжка и пр.).

Оперативные формы *web*-приложения представляют собой интерактивные элементы пользовательского интерфейса, которые позволяют пользователям взаимодействовать с *web*-приложением, отправлять данные и получать обратную связь в реальном времени без необходимости перезагрузки страницы. Примеры оперативных форм проектируемого *web*-приложения:

- форма авторизации;
- форма регистрации нового пользователя;
- форма для работы сотрудника деканата;
- форма для работы специалиста кафедры;
- форма для работы секретаря ГЭК;
- форма для работы преподавателя.

Виды выходных форм и отчетов:

- учебная карточка студента;
- электронный аналог зачетной книжки;
- протокол защиты в ГЭК;
- выписка из зачетно-экзаменационных ведомостей;
- приказ об отчислении по окончании обучения.

Для того, чтобы четко определить, какие функции должны быть реализованы в приложении и как оно должно взаимодействовать с пользователями необходимо определить его функциональные требования. Для проектируемого *web*-приложение были сформулированы следующие функциональные требования:

- регистрация нового пользователя и авторизация;
- обработка приказов о зачислении (отчислении), приказов о закреплении тем дипломных работ;
- ведение электронного аналога зачетной книжки;
- создание и редактирование учебной карточки студента;
- управление выписками из зачетно-экзаменационных ведомостей по дисциплинам;
- формирование итоговой документации о результатах учебной деятельности студентов (выписка из зачетно-экзаменационных ведомостей, протоколы защиты дипломных работ в ГЭК).

В связи с особенностями проектируемого *web*-приложения было принято решение использовать стек технологий *SERN*, который включает в себя *Express.js*, *React.js* и *Node.js*. Этот стек предоставляет гибкую и эффективную основу для разработки современных *web*-приложений.

Node.js выполняет функцию серверной среды выполнения, принимая запросы от клиентов и управляя взаимодействием с базой данных. *Express.js*, в свою очередь, предоставляет удобный фреймворк для *Node.js*, упрощая создание серверных приложений и обработку *HTTP*-запросов. *React.js* – это мощная *JavaScript*-библиотека для построения пользовательских интерфейсов. Она работает на стороне клиента и отвечает за отображение интерфейса и взаимодействие с сервером через *API*.

Когда пользователь взаимодействует с *web*-приложением, *React* отправляет запросы на сервер *Node.js* через *API*. *Express* обрабатывает эти запросы и, при необходимости, взаимодействует с базой данных. Затем сервер отправляет обратно в *React* соответствующие ответы, которые отображаются пользователю. Наглядный пример работы *SERN* представлен на рис. 1.

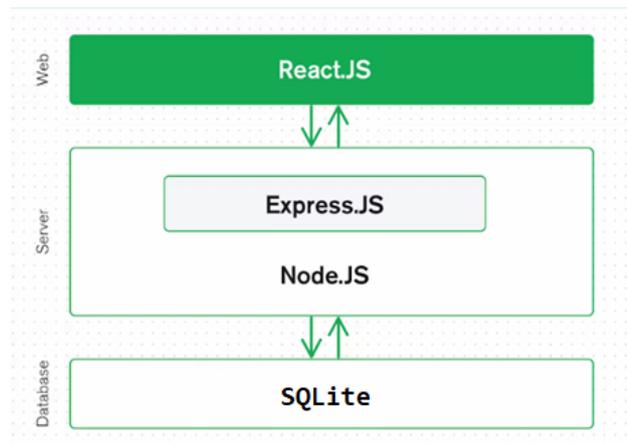


Рис. 1. Архитектура приложения

Так как *web*-приложение по учету учебной деятельности студентов ГГТУ им. П.О. Сухого предполагает взаимодействие с уже существующей базой данных, необходимо обеспечить ее гармоничную интеграцию с *web*-приложением. Чтобы сделать *JavaScript*-приложение не зависимым от конкретной базы данных будет использоваться так называемый «слой абстракции баз данных» или *ORM (Object-Relational Mapping)* при работе с реляционной базой данных. В качестве *ORM* для данного *web*-приложения был выбран *Sequelize*. Он предоставляет абстракцию над базой данных, позволяя работать с объектами *JavaScript* вместо запросов к базе данных напрямую.

Взаимодействие между клиентской и серверной частью происходит через *API*. *API*-интерфейсы *RESTful* служат мостом между интерфейсом и сервером в приложениях *SERN*. Они облегчают передачу данных, поддерживают архитектуру без сохранения состояния и обеспечивают масштабируемую разработку. *REST API* реализуется с помощью протокола *HTTP*, который используется для получения различных ресурсов в интернете.

REST – это архитектурный стиль, используемый при проектировании *web*-сервисов, который определяет правила для построения распределенных систем. Он основан на нескольких ключевых принципах:

- каждый компонент системы рассматривается как ресурс, доступ к которому осуществляется через уникальный идентификатор *URI*;
- *HTTP*-методы (*GET*, *POST*, *PUT*, *DELETE*) используются для выполнения операций над ресурсами;
- ресурсы могут быть представлены в различных форматах, таких как *JSON*, *XML* или *HTML*, в зависимости от запросов клиента.

Принцип работы *REST* состоит в том, что клиенты отправляют запросы на сервер с помощью *HTTP*-методов, указывая *URI* ресурса и требуемое действие. Сервер обрабатывает эти запросы и возвращает соответствующий результат, который может быть представлен в различных форматах. Наглядный пример представлен на рис. 2.

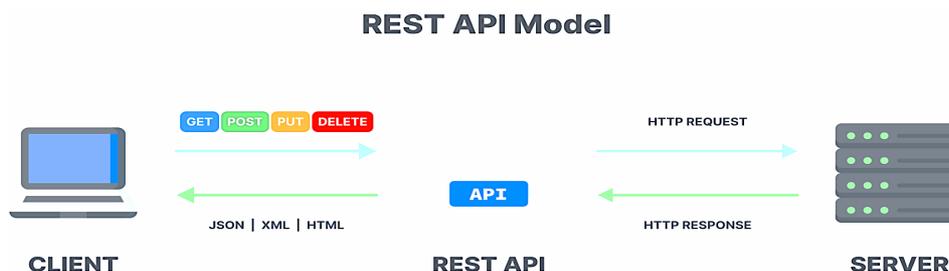


Рис. 2. Схема работы RESTAPI

Заключение. Таким образом, внедрение в учебную среду проектируемого *web*-приложения обеспечит эффективный учет и анализ учебной деятельности студентов, позволит автоматизировать процесс формирования отчетов и аналитики по результатам обучения, предоставит возможность упрощенного доступа к работе с необходимыми документами преподавателям, специалистам деканатов и администрации университета, сопровождающих учебный процесс.

Литература

1. MongoDB / «HowtoUseMERNStack: ACompleteGuide». – 2024. – Режим доступа: <https://www.mongodb.com/languages/mern-stack-tutorial>. – Дата доступа: 12.03.2024.
2. REST Framework / «Создание RESTful API с помощью Express и Node.js в стеке MERN». – 2024. – Режим доступа: <https://medium.com/@ibrahimhz/building-a-restful-api-with-express-and-node-js-in-mern-stack-e3714f94d24b> – Дата доступа: 12.03.2024.

РАЗРАБОТКА СЮЖЕТНОГО ИГРОВОГО ПРИЛОЖЕНИЯ В ЖАНРЕ «КОНЕЧНЫЙ РАННЕР» С ИСПОЛЬЗОВАНИЕМ АВТОРСКОЙ ГРАФИКИ В СРЕДЕ UNITY

О. В. Овчинина

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

Научный руководитель В. С. Захаренко

Рассмотрены основные проблемы при разработке игрового приложения «Echo-10» и их решения, такие как выбор средств разработки, архитектуры, процесс создания авторской графики, механик игры и сюжета.

Ключевые слова: игровое приложение, архитектура, графика, сюжет, разработка.

Игры в жанре «Раннер» давно укрепились в мире видеоигр. Они характеризуются процедурной генерацией уровней, созданием препятствий и системой подсчета очков. Под жанр «Конечный раннер» включает в себя наличие нескольких уровней и завершающий этап игры, что обеспечивает более детальное развитие сюжета игры.

Средства разработки. Для разработки игры был выбран язык программирования C#, игровой движок *Unity*, интегрированная среда разработки *Rider*, графический редактор *Adobe Photoshop* для создания оригинальной графики и сервис по управлению проектами *Notion*. Благодаря кросс-сплатформенной среде разработки *Unity* приложение адаптируется как для персонального компьютера, так и для мобильного устройства.