

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 21.Б10-мм

Автоматизация процессов подготовки и проведения защит ВКР

Мясникова Мария Геннадьевна

Отчёт по учебной практике
в форме «Производственное задание»

Научный руководитель:
старший преподаватель кафедры СП, к.т.н. Ю. В. Литвинов

Санкт-Петербург
2024

Оглавление

Введение	3
1. Постановка задачи	5
2. Требования	6
2.1. Роль администратора	6
2.2. Оценки	7
2.3. Голосование	7
2.4. Результаты защиты	7
3. Обзор используемых технологий	9
3.1. ASP.NET Core	9
3.2. Entity Framework Core	9
3.3. React	10
4. Реализация	11
4.1. Компонент Data	13
4.2. Компонент API	13
4.3. Компонент Client	15
5. Эксперимент	19
Заключение	21
Список литературы	22

Введение

Выпускная квалификационная работа (ВКР) – итог образовательного процесса, в котором студент демонстрирует специализированные навыки, приобретенные им в процессе освоения учебной программы. Каждая такая работа подлежит рецензированию и защите на заседании Государственной экзаменационной комиссии (ГЭК).

Координация ГЭК представляет собой длительный и трудоемкий бизнес-процесс, состоящий из большого количества этапов. В прошлом семестре один из них был автоматизирован при помощи разработанного консольного инструмента. Приложение позволило упростить работу с папками заседаний на Яндекс Диске, а также автоматически генерировать порядки дня для широкой публики и ГЭК.

Целью защиты ВКР являются определение уровня подготовленности выпускников и проверка сформированности навыков и умений, предусмотренных учебным планом образовательной программы. В ходе заседания члены комиссии слушают выступление защищающегося, отзывы научного руководителя и рецензента, а также задают вопросы касательно работы и отзывов. По окончании заседания члены ГЭК совещаются и при помощи критериев оценивают работу на основании ее содержания и оформления, доклада студента, отзыва и рецензии. Затем оценки усредняются и выставляется итоговая оценка ВКР, которая заносится в протокол защиты [9].

В рамках весеннего семестра предлагается начать работу над веб-приложением, позволяющим членам ГЭК заносить оценки в оценочный лист вместо заполнения таблицы на Яндекс Диске, так как офисный пакет Р7-Офис¹, используемый в качестве веб-редактора документов на Яндекс Диске, имеет ряд недостатков: тяжело использовать на мобильных устройствах, приходится долго ждать загрузки таблицы, возникают проблемы с синхронизацией изменений. Таким образом, необходимо разработать инструмент, предоставляющий членам комиссии всю необходимую информацию о заседании с возможностью оценить

¹Р7-Офис – российский офисный пакет. URL: <https://r7-office.ru> (дата обращения: 2024-10-02)

ВКР по критериям и автоматически подсчитать итоговую оценку работы. Предполагается, что данное веб-приложение также можно будет использовать на защите учебных практик.

1. Постановка задачи

Целью работы является разработка веб-приложения для оценивания ВКР членами комиссии. Для ее выполнения в весеннем семестре были поставлены следующие задачи:

1. Определить требования к приложению.
2. Провести обзор используемых технологий.
3. Спроектировать архитектуру приложения и определить структуру базы данных.
4. Разработать MVP-версию приложения.
5. Протестировать разработанный инструмент.

2. Требования

Требуется разработать веб-приложение для оценивания ВКР членами ГЭК. Диаграмма случаев использования приложения представлена на Рис. 1

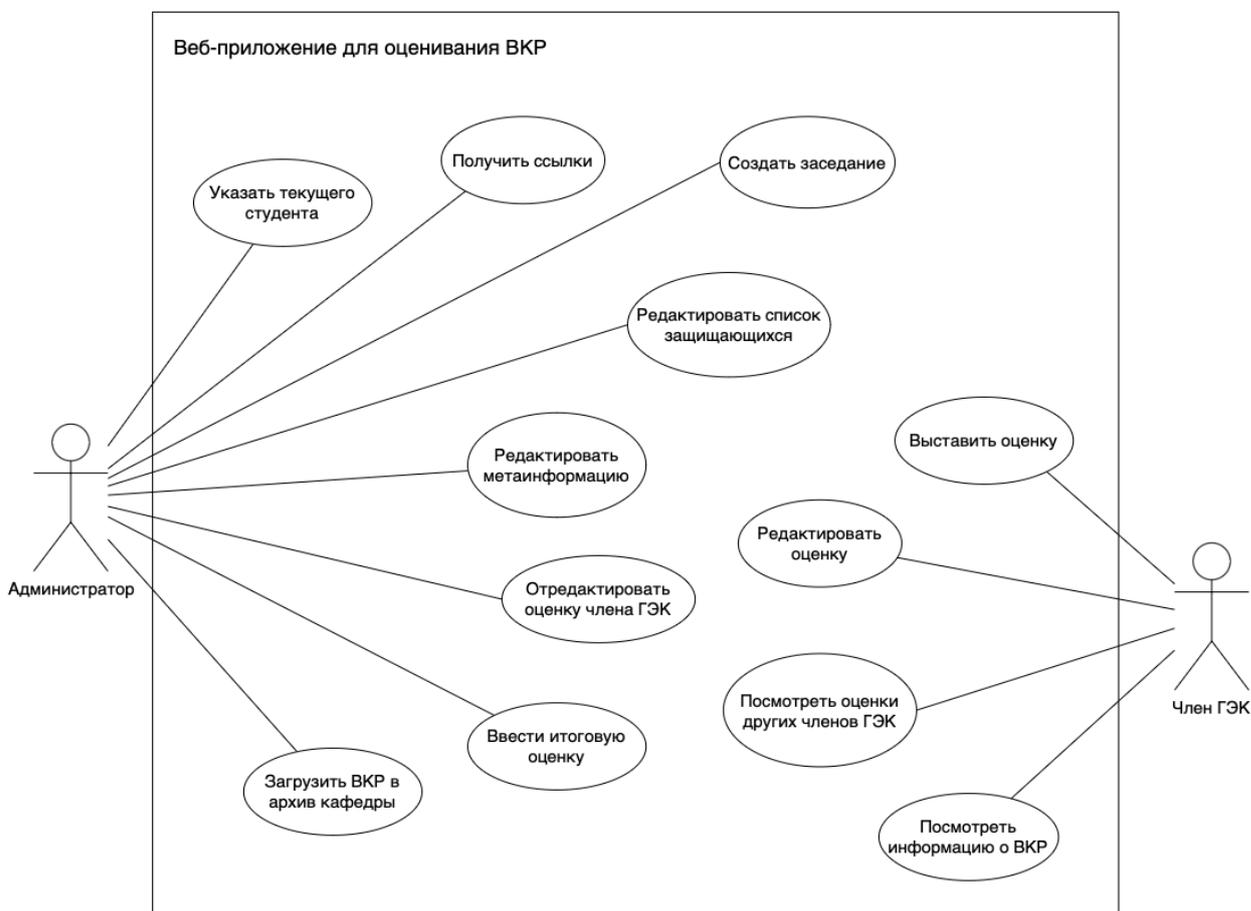


Рис. 1: Диаграмма случаев использования веб-приложения

2.1. Роль администратора

В приложении должна присутствовать роль администратора, который может «создавать» заседание, задавать списки защищающихся, указывать ссылку на видеоконференцию для обсуждения ВКР, а также редактировать любую метаинформацию. Данные, касающиеся студентов и их работ, могут быть получены из .xlsx-таблиц, хранящихся на Яндекс Диске. В результате генерируется ссылка на заседание, которую

администратор распространяет членам комиссии. Возможна также генерация ссылки для общего доступа.

Для удобства членов ГЭК в процессе защиты администратор может указать текущего защищаемого студента.

2.2. Оценки

В ходе защиты каждый член комиссии выставляет оценки по критериям текущему защищаемому. Каждый критерий состоит из:

- названия критерия;
- комментария к критерию;
- шкалы оценивания;
- дополнительных правил (например, «отсутствие СИ в репозитории даёт -2 к баллам за реализацию»).

Члены комиссии могут менять свою оценку вплоть до окончания заседания и видеть оценки, поставленные другими участниками. Администратор, в свою очередь, имеет право редактировать любую оценку.

2.3. Голосование

После защиты администратору и всем членам комиссии должны быть видны усреднённые оценки и итоговая оценка за работу для обсуждения. На этапе голосования оценки всё ещё можно редактировать.

Посчитанная оценка имеет рекомендательное значение для комиссии, так как итоговая оценка может быть изменена в результате обсуждения.

2.4. Результаты защиты

По результатам защиты должно происходить следующее:

1. Формируется .xlsx-таблица со всеми оценками, выставленными членами комиссии, средними оценками и итоговой оценкой.
2. Генерируется ссылка для студентов, по которой они могут ознакомиться с результатами защиты (средними оценками по критериям и итоговой оценкой после обсуждения).
3. Успешно защищённые работы по команде администратора загружаются в архив кафедры.

3. Обзор используемых технологий

В силу специфичности разрабатываемого сервиса обзор решений-аналогов не проводился.

При выборе языка программирования важно было учесть, что большая часть существующих решений, связанных с автоматизацией задач университета, использует платформу .NET [5]. Таким образом, для разработки веб-приложения был выбран C#, как широко используемый объектно-ориентированный язык для этой платформы [3].

3.1. ASP.NET Core

ASP.NET Core [1] — кроссплатформенный и высокопроизводительный фреймворк для разработки веб-приложений и API. В качестве подхода к разработке веб-сервиса используется Minimal API. В отличие от традиционной MVC-архитектуры, Minimal API не требует создания контроллеров и моделей, что ускоряет процесс написания кода, и, несмотря на простоту, полностью совместим с экосистемой ASP.NET Core.

3.2. Entity Framework Core

Entity Framework Core [4] — кроссплатформенная ORM-система (Object-Relational Mapping) для .NET, которая упрощает работу с реляционными базами данных, позволяя взаимодействовать с ними через объекты вместо написания SQL-запросов. В проекте используется PostgreSQL [6] — объектно-реляционная система управления базами данных, поддерживающая транзакции, масштабируемость и сложные запросы. Entity Framework Core предоставляет инструменты для работы с данной СУБД, включая управление схемой базы данных через миграции и выполнение запросов с использованием LINQ. Выбор PostgreSQL позволил получить опыт работы с одной из самых востребованных систем управления базами данных.

3.3. React

React [7] — библиотека для создания интерактивных пользовательских интерфейсов, основанная на компонентном подходе. Она позволяет эффективно обновлять интерфейс с помощью виртуального DOM (Document Object Model) и поддерживает создание адаптивных веб-приложений, которые корректно отображаются на различных устройствах, включая мобильные. React широко используется в сообществе разработчиков, что обеспечивает обширную базу знаний и примеров. Выбор обусловлен также стремлением получить опыт работы с этой популярной библиотекой. Для сборки фронтенда используется Vite [8] — современный инструмент, обеспечивающий быструю перезагрузку и оптимизированный процесс разработки. Для работы с API используется Axios [2] — популярная библиотека для отправки HTTP-запросов, которая облегчает обработку асинхронных операций и работу с запросами и ответами от сервера.

4. Реализация

Перед началом реализации была спроектирована архитектура будущего веб-приложения (Рис. 2).

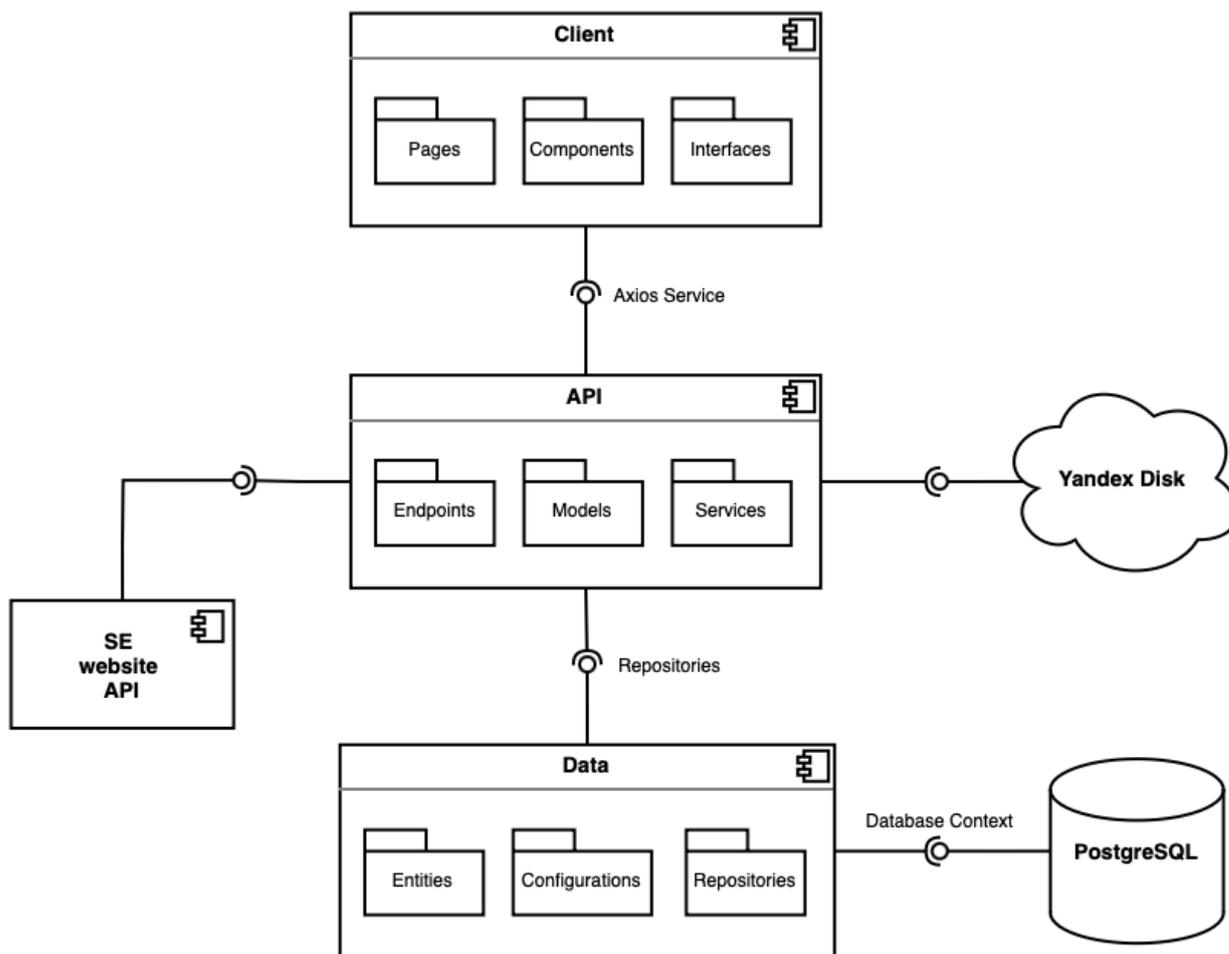


Рис. 2: Диаграмма компонентов веб-приложения

Приложение состоит из двух основных частей: клиентской и серверной. Клиентская часть представляет собой интерфейс пользователя и сервис для взаимодействия с сервером – Axios Service. Серверную часть образуют компоненты API и Data. Подробное описание компонентов приведено ниже.

Также была определена структура базы данных (Рис. 3). Основные сущности:

- User – пользователь с указанием роли (Role). На данный момент определены две роли: администратор и член комиссии. В случае

роли администратора хранит пароль для входа, в случае роли члена комиссии – идентификатор заседания.

- Meeting – заседание комиссии, имеет связь один ко многим с сущностью StudentWork (работа студента).
- Criteria – критерий, имеет связи многие ко многим с таблицей заседаний и один ко многим с таблицей правил (Rule). Правила могут относиться к шкале оценивания или являться дополнительными. Для хранения оценок участников комиссии по критериям определена сущность MemberMark. Таблица CriteriaMark представляет собой усредненную оценку работы по критерию.

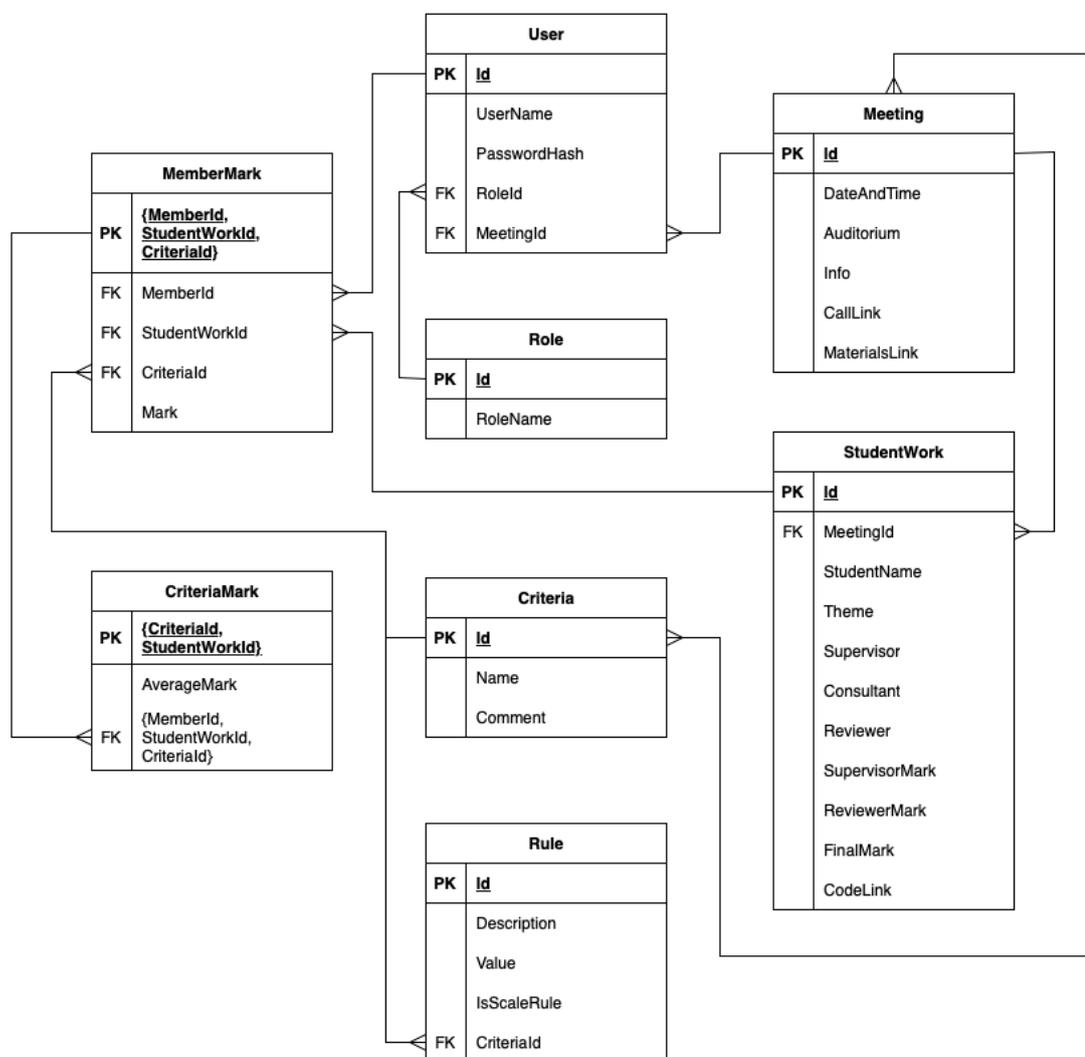


Рис. 3: ER-диаграмма базы данных

4.1. Компонент Data

Данный компонент отвечает за организацию и управление данными в приложении. Он включает несколько ключевых частей, которые обеспечивают корректную работу с базой данных:

- Сущности – классы, описывающие объекты базы данных.
- Конфигурации – классы, используемые для определения параметров сущностей, таких как ключи, ограничения, связи и начальные данные.
- Репозитории – классы для обеспечения доступа к данным и инкапсуляции логики работы с базой, предоставляющие простые методы для выполнения операций с сущностями.

Центральным объектом для взаимодействия с базой данных является контекст. Он отслеживает изменения в сущностях, выполняет запросы к базе и управляет транзакциями.

4.2. Компонент API

Данный компонент приложения отвечает за обработку поступающих HTTP-запросов и взаимодействие с компонентом данных Data, которое осуществляется при помощи сервисов – UserService, MeetingService и CriteriaService, каждый из которых использует соответствующие репозитории.

Для реализации авторизации и аутентификации пользователей в приложении используются JWT-токены (JSON Web Tokens). Токен содержит зашифрованную информацию о пользователе, включая его имя, роль и срок действия токена. Валидация осуществляется с использованием секретного ключа, хранящегося на сервере. Если токен недействителен или срок его действия истёк, доступ к защищённым маршрутам блокируется. Благодаря хранению в токене роли пользователя стало возможным использование запрещающих политик для ограничения доступа к администраторской части сервиса.

Для структурирования и валидации данных, поступающих от клиента и передаваемых сервером, используются модели запросов и объекты передачи данных (DTO). Они обеспечивают безопасность и корректность данных, а также помогают в их преобразовании между различными слоями приложения.

API содержит набор маршрутов (endpoints), которые обеспечивают доступ к функциональности приложения. Эндпоинты разделены на три основные группы по типу данных, с которыми происходит взаимодействие: маршруты критериев, маршруты заседаний и маршруты пользователей. На данный момент единственным эндпоинтом, относящимся к пользователям, является путь «/login» для входа администратора (Рис. 4).

CriteriaEndpoints	
POST	/criteria/new
GET	/criteria
PUT	/criteria/update
DELETE	/criteria/delete

MeetingEndpoints	
POST	/meetings/new
GET	/meetings
PUT	/meetings/update
DELETE	/meetings/delete

PracticeGrading.API	
POST	/login

Рис. 4: Маршруты API

На диаграмме компонентов также отражены сервисы, с которыми веб-приложение будет взаимодействовать в будущем – API сайта кафедры Системного Программирования² и REST API Яндекс Диска³, так как защищенные работы должны будут загружаться в архив ка-

²Кафедра системного программирования СПбГУ. URL: <https://se.math.spbu.ru> (дата обращения: 2024-11-19)

³REST API Диска — Технологии Яндекса. URL: <https://yandex.ru/dev/disk/rest> (дата обращения: 2024-11-19)

федры, а при создании заседаний некоторую информацию можно будет получить из таблиц, хранящихся на Диске.

4.3. Компонент Client

Данный компонент относится к клиентской части веб-приложения и предоставляет пользователю интерфейс для взаимодействия с сервером. Первой была реализована страница входа для администратора (Рис. 5). После успешной авторизации пользователь получает токен, который в будущем будет использоваться Axios Service'ом в запросах к API для контроля доступа.

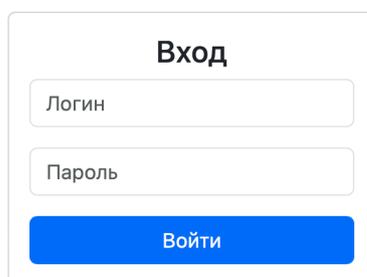


Рис. 5: Страница входа

Главная страница веб-приложения – «Заседания», отображающая все созданные заседания (Рис. 6). С помощью кнопок администратор может просмотреть или удалить выбранное заседание, а также создать новое.

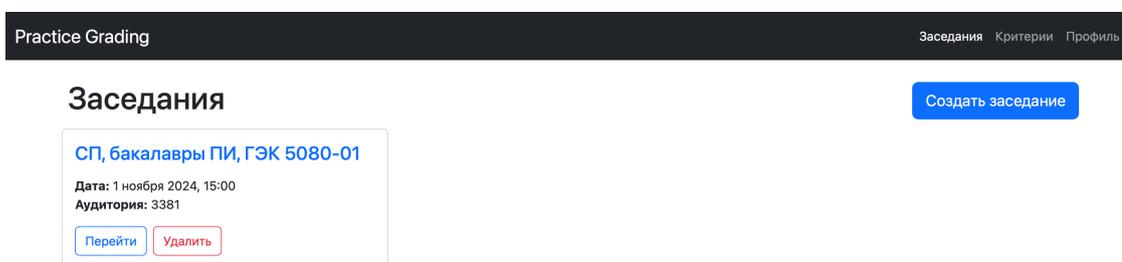


Рис. 6: Главная страница со списком заседаний

Practice Grading Заседания Критерии Профиль

Новое заседание Сохранить Назад

Дата и время: 20 ноября 2024, 00:03

Аудитория: 3381

Информация о заседании: СП, бакалавры ПИ, ГЭК 5080-01

Ссылка на созвоп: https://...

Ссылка на материалы: https://...

Список защищающихся студентов Добавить студента

ФИО	Тема	Научник	Консультант	Рецензент	Оценка научника	Оценка рецензента	Код
Студент	Тема	Научник	—	—	4	5	—

Список членов комиссии: Иванов Иван Иванович

Критерии

- Критерий 1
комментарий
- Критерий 2
- Критерий 3

Рис. 7: Страница создания заседания

Страница создания представлена на Рис. 7. Помимо общей информации, пользователь может добавлять, редактировать и удалять студенческие работы (Рис. 8), указывать членов комиссии и критерии оценивания.

СП, бакалавры ПИ, ГЭК 5080-01

Создание студенческой работы ✕

ФИО студента:

Тема практики/ВКР:

Научный руководитель:

Консультант:

Рецензент:

Ссылка на код:

Оценка научного руководителя: Оценка рецензента:

Рис. 8: Окно создания работы студента

При нажатии на кнопку «Перейти» у заседания администратор попадает на страницу просмотра заседания, где может перейти к редактированию (Рис. 9).

Practice Grading Заседания Критерии Профиль

Просмотр заседания

[Редактировать](#) [Назад](#)

Дата и время 1 ноября 2024, 15:00

Аудитория 3381

Информация о заседании СП, бакалавры ПИ, ГЭК 5080-01

Список защищающихся студентов

ФИО	Тема	Научник	Консультант	Рецензент	Оценка научника	Оценка рецензента	Код
Иванов Иван Иванович	Разработка системы оценки эффективности учебных мероприятий	Петров Петр Петрович	Сидорова Мария Владимировна	Кузнецова Ольга Сергеевна	4	5	—
Смирнова Анна Владимировна	Внедрение технологий искусственного интеллекта в образовательный процесс	Васильев Алексей Николаевич	—	Горбунов Дмитрий Сергеевич	5	—	Ссылка
Кузнецов Артем Сергеевич	Анализ влияния цифровых инструментов на продуктивность студентов	Захаров Михаил Александрович	—	—	—	—	—

Список членов комиссии

Лебедев Сергей Владимирович

Федорова Елена Васильевна

Критерии

Критерий 1 комментарий
Критерий 2
Критерий 3

Рис. 9: Страница просмотра заседания

При помощи навигационной панели пользователь также может открыть страницу «Критерии» и взаимодействовать с ними – изменять и удалять их из списка (Рис. 10). Кнопка «Добавить критерий» открывает модальное окно для создания нового критерия (Рис. 11).

Список критериев

Добавить критерий

Критерий 1 комментарий	^ × ✎
Шкала оценивания	
5 — отлично	
4 — хорошо	
3 — удовлетворительно	
2 — неудовлетворительно	
Дополнительные правила	
-1 ничего не работает	
Критерий 2	∨ × ✎
Критерий 3	∨ × ✎

Рис. 10: Страница со списком критериев

Создание критерия

Название

Комментарий

Шкала оценивания

5	здорово
0	Оставьте пустым, если не хотите добавлять это правило

Дополнительные правила

-1	так себе
0	Оставьте пустым, если не хотите добавлять это правило

Отмена Сохранить

Рис. 11: Окно создания критерия

5. Эксперимент

На данный момент проводилось только ручное тестирование разработанного инструмента. Были проверены следующие тестовые сценарии:

- **Вход в приложение:** при вводе корректного логина и пароля авторизация успешно выполняется, в сессионном хранилище появляется токен. В случае неверных данных показывается предупреждение об ошибке.
- **Совершение действий в приложении с истекшим токеном:** при переходе между страницами и их обновлении в момент истечения срока жизни токена пользователь перенаправляется на страницу входа, токен удаляется из хранилища.
- **Выход из приложения:** при нажатии на кнопку «Выйти» пользователь перенаправляется на страницу входа, токен удаляется из хранилища.
- **Создание заседания:** при указании необходимых полей (информация о заседании и хотя бы одна работа студента) заседание сохраняется, пользователь перенаправляется на главную страницу, где отображается созданное заседание. В базе данных появляются соответствующие записи: новое заседание, созданные работы, указанные члены комиссии, а также связь с выбранными критериями.
- **Удаление заседания:** пользователь видит предупреждение, подтверждает намерение удалить заседание, после чего оно исчезает из списка. Из базы данных удаляются записи об этом заседании, работах студентов, членах комиссии и связь с критериями.
- **Редактирование заседания:** текущие данные отображаются корректно, изменения сохраняются в базе, пользователь возвращается к просмотру заседания.

- Аналогичным образом были протестированы сценарии создания, редактирования и удаления критериев и студенческих работ.

Заключение

В рамках учебной практики за весенний семестр были достигнуты следующие результаты:

- Определены требования к веб-приложению.
- Проведен обзор используемых технологий.
- Спроектирована архитектура приложения и определена структура базы данных.
- Реализованы следующие возможности:
 - Авторизация и аутентификация пользователей.
 - Создание, просмотр, редактирование и удаление заседаний.
 - Создание, просмотр, редактирование и удаление критериев.
- Реализованы соответствующие страницы и модальные окна.
- Проведено ручное тестирование разработанного веб-приложения.

Реализация доступна на GИTНUB⁴ (имя аккаунта: mawekk).

⁴<https://github.com/yurii-litvinov/PracticeGrading>

Список литературы

- [1] ASP.NET Core. — URL: <https://dotnet.microsoft.com/ru-ru/apps/aspnet> (дата обращения: 2024-11-18).
- [2] Axios. — URL: <https://axios-http.com> (дата обращения: 2024-11-18).
- [3] C#. — URL: <https://dotnet.microsoft.com/ru-ru/languages/csharp> (дата обращения: 2024-08-18).
- [4] Entity Framework Core. — URL: <https://learn.microsoft.com/en-us/ef/core/> (дата обращения: 2024-11-18).
- [5] .NET. — URL: <https://dotnet.microsoft.com/ru-ru/> (дата обращения: 2024-08-18).
- [6] PostgreSQL. — URL: <https://www.postgresql.org> (дата обращения: 2024-11-18).
- [7] React. — URL: <https://ru.react.dev> (дата обращения: 2024-11-18).
- [8] Vite. — URL: <https://vite.dev> (дата обращения: 2024-11-18).
- [9] Приказ № 14453/1 от 13.11.2023 «О методическом обеспечении государственной итоговой аттестации в 2024 году (СВ.5162.*)». — URL: https://edu.spbu.ru/files/2023/20231113_14453_1.pdf (дата обращения: 2024-10-02).