Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 22.Б11-мм

Расширение функциональности по приглашению студентов в сервисе HwProj

Аверин Павел Владиславович

Отчёт по производственной практике
в форме «Решение»

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

Консультант:
инженер-программист «JetBrains» А. В. Бережных

Санкт-Петербург
2025

Оглавление

Введение **3**

1. Постановка задачи **4**

2. Обзор **5**

2.1. Хранение и извлечение информации о студентах 5

2.2. Механизмы регистрации студентов 6

2.3. Используемые технологии 6

2.4. Выбор библиотеки для работы с Active Directory и LDAP 7

2.5. Архитектура HwProj 7

2.6. Работа за осенний семестр 9

3. Реализация **12**

3.1. Бэкенд 12

3.2. Фронтенд 12

Апробация **14**

Заключение **15**

Список литературы **16**

Введение

В условиях стремительного прогресса цифровых технологий их интеграция в образовательную сферу приобретает ключевое значение. Учебные заведения всё чаще внедряют специализированные онлайн-платформы и инструменты, направленные на автоматизацию преподавания, усиление коммуникации между участниками образовательного процесса и обеспечение высоких стандартов обучения. Особую актуальность в данном контексте приобретают системы управления курсами и мониторинга академических достижений студентов.

Среди подобных платформ можно выделить HwProj [5] — систему, активно применяемую в СПбГУ, а также в других вузах. Сервис специализируется на организации учебных курсов в сфере IT, компьютерных наук, программирования и смежных областей.

У данного проекта, несмотря на активную разработку и поддержку, имелся существенный недостаток, который отсутствовал в аналогичных сервисах, таких как Blackboard [2] и MS Teams [7], — отсутствие интеграции с университетской базой данных, что усложняло процесс приглашения студентов на курс. Целью осенней учебной практики стала интеграция университетской базы данных с сервисом. В ходе работы был создан пул-реквест в основной репозиторий проекта [18], который после доработок в период весенней практики был принят.

Кроме того, в рамках данной работы была поставлена задача **реализовать механизм добавления студентов на курс преподавателем**, что логически продолжало работу, начатую в ходе производственной практики в предыдущем семестре как часть цели по **расширению функциональности по приглашению студентов в HwProj**.

1. Постановка задачи

Целью работы является расширение функциональности по приглашению студентов в сервисе HwProj. Для её выполнения были поставлены следующие задачи:

1. Завершить работу, которая была начата в осеннем семестре.
2. Провести обзор того, как СПбГУ хранит информацию о студентах.
3. Спроектировать и реализовать функциональность по извлечению данных о студентах на стороне бэкэнда.
4. Спроектировать и реализовать функциональность по извлечению данных об образовательных программах и учебных группах на стороне бэкэнда.
5. Интегрировать реализованную функциональность с фронтендом.

2. Обзор

2.1. Хранение и извлечение информации о студентах

Санкт-Петербургский государственный университет использует Active Directory (AD) для управления данными студентов. AD — служба каталогов Microsoft, обеспечивающая централизованное управление сетевыми ресурсами и хранение пользовательской информации. AD использует протокол LDAP для организации доступа к данным. Этот протокол позволяет выполнять запросы к каталогу и находить пользовательские данные. Поэтому данные студентов решено извлекать из AD по LDAP.

Запись в LDAP имеет уникальное имя DN (Distinguished Name), которое состоит из атрибутов и их значений. Например, в DN "cn=John Smith, ou=Staff, dc=example, dc=com" каждый элемент называется RDN (Relative Distinguished Name) и имеет формат "ИмяАтрибута=значение".

Для работы с LDAP нужен аккаунт с правами поиска и извлечения записей. В качестве таких аккаунтов подойдут служебные аккаунты студентов, однако их использование нежелательно из-за того что они перестают быть действительными после окончания учёбы.

Active Directory не содержал информация об образовательной программе, информацию о них соответствующих им группах было изначально решено взять из сервиса электронного расписания Timetable [13], разбирая HTML страницу с помощью библиотеки AgilityPack [4], однако весной в ходе работы над пул-реквестом был найден API Timetable, на который было переписано извлечение нужной информации. Преимуществами данного подхода являются:

- **Надёжность**: API предоставляет структурированные данные в формате JSON/XML, что снижает риск ошибок из-за изменений в вёрстке HTML.
- **Стабильность**: API менее подвержен изменениям, чем HTML-структура, поэтому код требует меньше поддержки.
- **Масштабируемость**: API позволяет гибко выбирать нужные данные, уменьшая объём передаваемой информации.
- **Производительность**: Запросы к API обычно работают быстрее, чем загрузка и парсинг всей HTML-страницы.
- **Поддержка версионности**: Многие API имеют версии, что позволяет контролировать изменения и избегать внезапных поломок.

2.2. Механизмы регистрации студентов

Изначально для регистрации студентов рассматривался стандартный вариант с отправкой ссылки для самостоятельного создания учётной записи. Этот подход прост в реализации, но не подходит для срочных процессов вроде переводов или восстановления, где важна скорость.

Альтернативный вариант — генерация токена и отправка специальной ссылки для входа. Такой механизм позволяет сразу предоставить доступ без ручного создания аккаунта. Дополнительный плюс — возможность сделать аккаунты временными, что полезно для определённых сценариев. Подобный метод уже используется в системе для приглашения экспертов.

Оба подхода имеют свои плюсы: первый требует меньше усилий для внедрения, второй обеспечивает гибкость и быстроту. Выбор зависит от конкретных задач — например, для массовой регистрации подойдёт первый вариант, а для экстренных случаев — второй.

2.3. Используемые технологии

Большинство технологий уже использовалось в проекте до этого и как следствие не менялось:

- **C#** [3] — главный язык программирования на стороне бэкенда, который используется в проекте.
- **RabbitMQ** [9] — брокер сообщений с открытым исходным кодом.
- **Swagger** [11] — набор инструментов для проектирования, документирования и тестирования API.
- **Material UI** [6] — библиотека компонентов для React.
- **React** [10] — это JavaScript-библиотека для создания пользовательских интерфейсов.
- **Typescript** [14] — главный язык программирования на стороне фронтенда, который используется в проекте.
- **ASP.NET Core** [1] — это кросс-платформенный фреймворк для создания веб-приложений, разработанный Microsoft.

2.4. Выбор библиотеки для работы с Active Directory и LDAP

В качестве библиотеки для работы с LDAP в прошлом семестре была выбрана Novell.Directory.Ldap [8], и этот выбор остался неизменным. В качестве альтернативы также рассматривалась библиотека System.DirectoryServices.Protocols [12], однако она оказалась менее подходящей для данной работы. Причина заключалась в том, что данная библиотека предоставляет низкоуровневый и излишне детализированный доступ к LDAP, что не требовалось в рамках проекта. Кроме того, библиотека от Novell обладала более качественной документацией и была проще в использовании, что и обусловило выбор в её пользу.

2.5. Архитектура HwProj

Фронтендная часть представляет собой React-приложение, а бэкенд построен на основе сервисов, разработанных с использованием ASP.NET.

Основные сервисы включают:

- **NotificationsService** — обрабатывает отправку уведомлений пользователям.
- **CoursesService** — управляет созданием курсов и операциями с ними.
- **SolutionsService** — отвечает за обработку решений, предоставляемых студентами.
- **AuthService** — обеспечивает процессы аутентификации и авторизации пользователей.

Коммуникация между клиентом и сервером происходит через «API Gateway», выполняющий роль единой точки входа и распределяющий запросы по соответствующим сервисам. Для межсервисного взаимодействия используется брокер сообщений RabbitMQ.

Архитектура изображена на Рисунке 1. позаимствованным из текста работы за осеннего семестра [19].

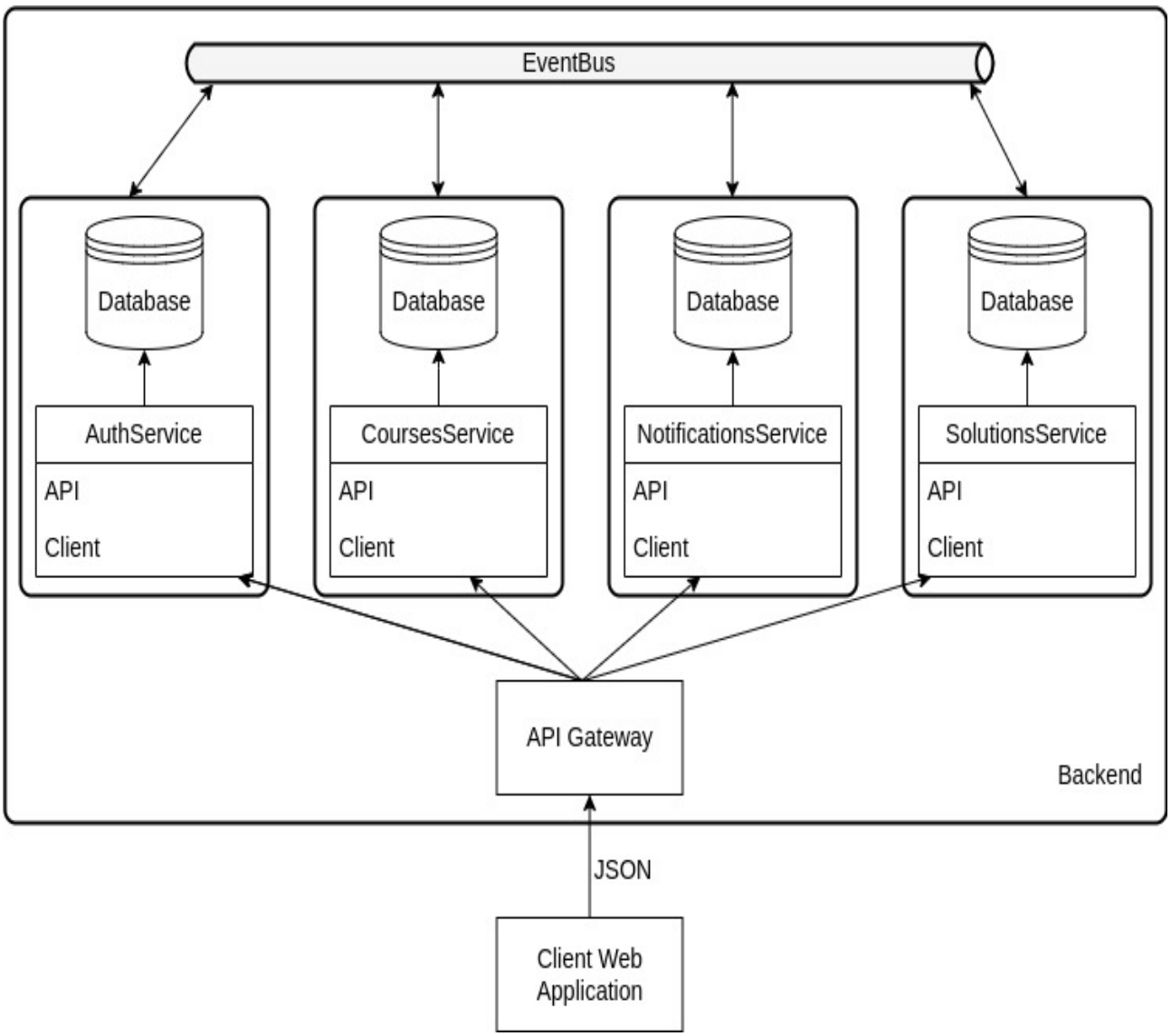


Рис. 1: Архитектура серверной части HwProj.

2.6. Работа за осенний семестр

В ходе весенней работы, реализация за осенний семестр не потерпела весомых изменений с архитектурной точки зрения.

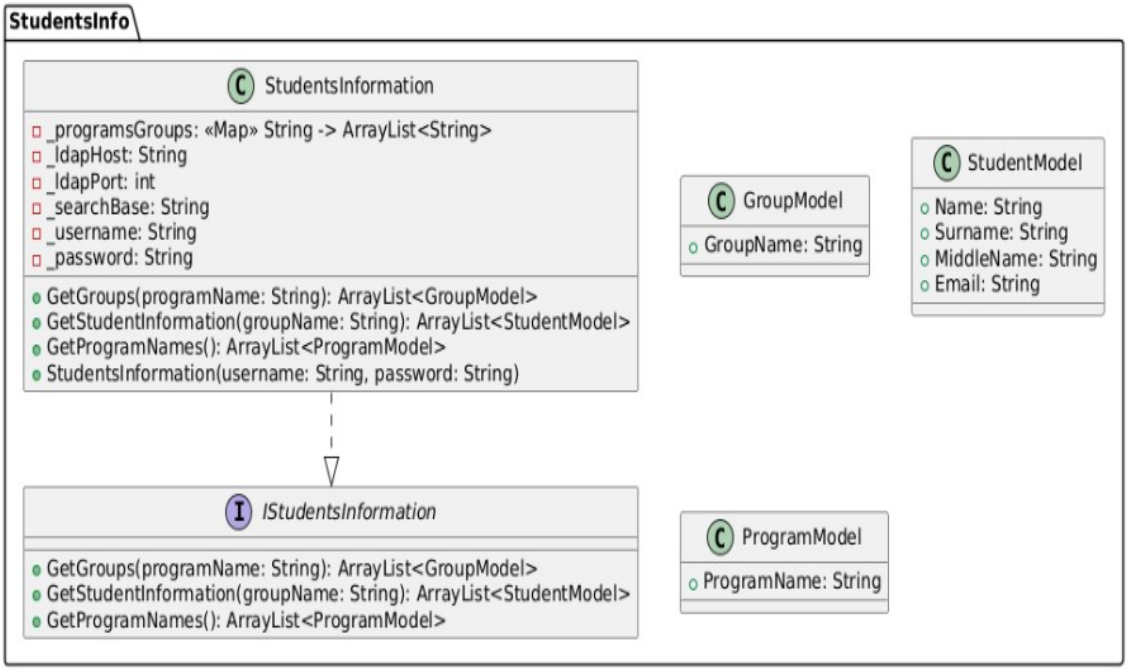


Рис. 2: UML-диаграмма класса StudentsInformation.

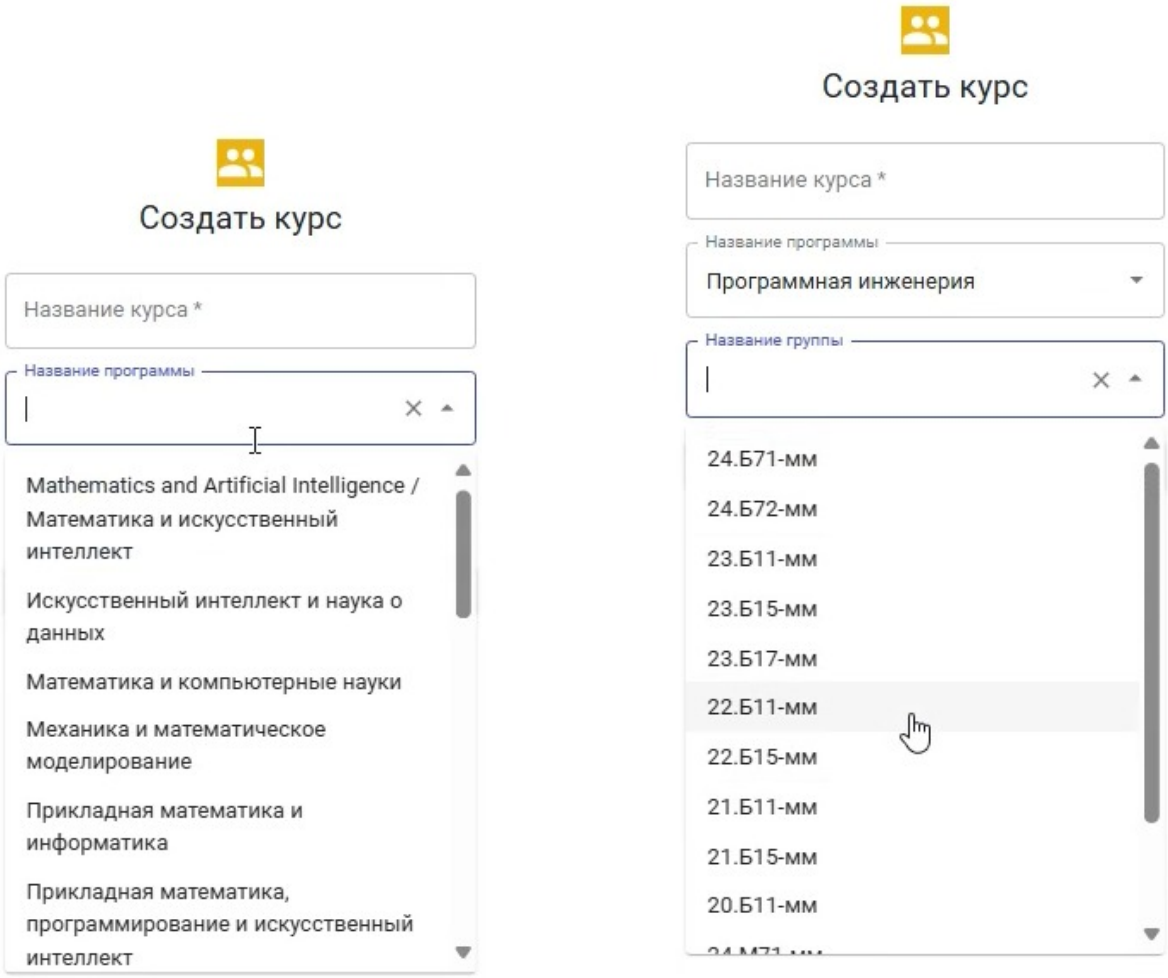
Был создан класс *StudentsInformation* (Рисунок 2. [19]) для работы с данными студентов, учебных групп и образовательных программ.

2.6.1. Извлечение данных о группах и программах

- Конструктор класса обращается к API Timetable, сохраняя названия программ *_programNames* и связанные группы *_programGroups*.
- Для получения списка программ используется свойство ProgramNames.
- Метод *GetGroups* возвращает группы по названию программы.
- На фронтенде добавлены компоненты Autocomplete (Material UI), работающие через контроллеры *GetProgramNames* и *GetGroups*.

2.6.2. Извлечение данных о студентах

- Метод *GetStudentInformation* подключается к Active Directory (LDAP) по логину/паролю, фильтрует записи по группе (*groupName*) и формирует словарь «почта → ФИО» (на основе атрибутов *cn* и *displayName*).
- Контроллер *InviteAndRegisterGroup* принимает группу и ID курса, регистрирует студентов (если нужно) и добавлет на курс.



(a) Автодополнение для имени образовательной программы.

(b) Автодополнение для учебной группы.

Рис. 3: Поля для названия программы и группы на странице создания курса. [19]

3. Реализация

3.1. Бэкенд

В ходе работы были реализованы следующие контроллеры на стороне бэкедна:

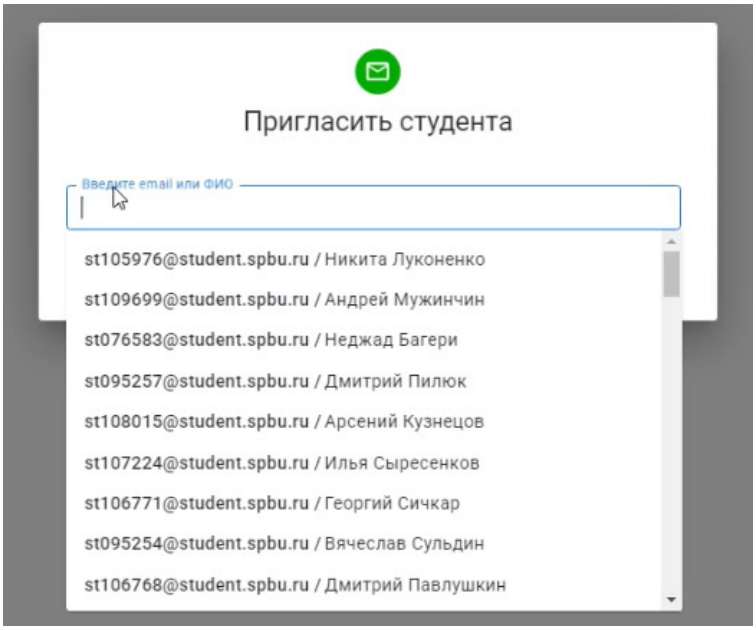
- **GetStudentToken** – GET-метод. Принимает email студента и возвращает токен для этого студента. Используется для генерации временного токена доступа.
- **LoginWithToken** – POST-метод, который позволяет аутентифицироваться по токену. Принимает учетные данные токена (*TokenCredentials)* и возвращает результат входа в систему.
- **ValidateToken** – POST-метод для проверки валидности токена. Принимает токен (*TokenCredentials*) и возвращает результат его проверки, включая информацию о его действительности.
- **InviteStudent** – POST-метод. Принимает модель *InviteStudentViewModel*, которая содержит данные студента (email, имя, фамилию и отчество) и идентификатор курса. Если студент не найден по email, он регистрируется в системе, после чего приглашается на указанный курс. В случае успеха возвращает подтверждение о приглашении студента.

3.2. Фронтенд

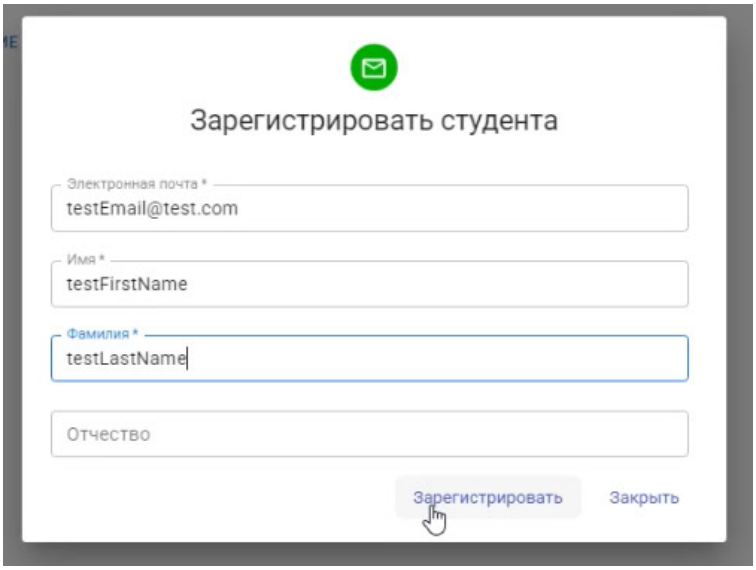
В ходе работы на фронтенде были разработаны следующие компоненты:

- **InviteStudentDialog** – Это диалоговое окно для приглашения студента на курс. Оно позволяет преподавателю:
    - Искать студента по почте или ФИО через *Autocomplete*.
    - Приглашать существующего студента, если он найден в системе.
    - Переходить к регистрации нового студента, если он не найден.

- **RegisterStudentDialog** – Это диалоговое окно для регистрации нового студента и его автоматического приглашения на курс. Оно позволяет преподавателю регистрация студента, отправляя данные (почту, имя, фамилия, отчество) на бэкенд. После чего он будет автоматически приглашен на курс.



(a) Диалоговое для приглашения зарегистрированных студентов на курс.



(b) Диалоговое для регистрации студентов и приглашения на курс.

Рис. 4: Разработанные диалоговые окна.

Апробация

В ходе работы были открыты и **приняты** два пул-реквеста, дополняющие и улучшающую функциональность, реализованную осенью:

- Пул-реквест, переписывающий разбор HTML страницы на использование API [16].
- Пул-реквест, позволяющий создавать курс с несколькими группами, а не только с одной. [17]

На момент написания текста пул-реквест с приглашением и регистрацией студентов был открыт в репозиторий проекта [15].

Была проведена частичная апробация со стороны преподавателей: есть положительные отзывы касательно функциональности по получению образовательных программ и групп. Однако провести полноценную апробацию, включающую получение данных о студентов из базы данных не удалось в виду того, что служебный аккаунт на момент написания текста так и не был предоставлен. Тем не менее, данная функциональность была локально протестирована.

Заключение

В ходе выполнения данной работы были достигнуты следующие результаты.

1. Была завершена начатая работа за осенний семестр, был принят основной пул-реквест [18] и два дополнительных в репозиторий проекта [16] [17].
2. Спроектирована и реализована функциональность по добавлению зарегистрированных на сервисе студентов на курс.
3. Спроектирована и реализована функциональность по добавлению незарегистрированных на сервисе студентов на курс.
4. Реализованная функциональность интегрирована с фронтендом.

Основной пул-реквест с приглашением студентов на курс можно посмотреть по ссылке [15].

Кроме всего прочего, работа заняла призовое место в конкурсе стипендий YADRO 2025

Список литературы

[1] ASP.NET Core. — (дата обращения: 12 июня 2025 г.). URL: https://dotnet.microsoft.com/en-us/apps/aspnet.

[2] Blackboard Learn. — (дата обращения: 12 июня 2025 г.). URL: https://bb.spbu.ru/.

[3] C#. — (дата обращения: 12 июня 2025 г.). URL: https://dotnet.microsoft.com/en-us/languages/csharp.

[4] Html Agility Pack. — (дата обращения: 12 июня 2025 г.). URL: https://html-agility-pack.net/.

[5] HwProj. — (дата обращения: 12 июня 2025 г.). URL: https://github.com/IteIIigeNET/HwProj-2.0.1.

[6] Material UI. — (дата обращения: 12 июня 2025 г.). URL: https://mui.com/material-ui/.

[7] Microsoft Teams. — (дата обращения: 12 июня 2025 г.). URL: https://www.microsoft.com/en-us/microsoft-teams/group-chat-software.

[8] Novell.Directory.Ldap.NETStandard. — (дата обращения: 12 июня 2025 г.). URL: https://github.com/dsbenghe/Novell.Directory.Ldap.NETStandard.

[9] RabbitMQ. — (дата обращения: 12 июня 2025 г.). URL: https://www.rabbitmq.com/.

[10] React. — (дата обращения: 12 июня 2025 г.). URL: https://react.dev/.

[11] Swagger. — (дата обращения: 12 июня 2025 г.). URL: https://swagger.io/.

[12] System.DirectoryServices.Protocols. — (дата обращения: 12 июня 2025 г.). URL: https://www.nuget.org/packages/System.DirectoryServices.Protocols/.

[13] Timetable. — (дата обращения: 12 июня 2025 г.). URL: https://timetable.spbu.ru/.

[14] Typescript. — (дата обращения: 12 июня 2025 г.). URL: https://www.typescriptlang.org/.

[15] Основной пул-реквест за весеннюю практику. — (дата обращения: 12 июня 2025 г.). URL: https://github.com/IntelligeNET/HwProj-2.0.1/pull/587.

[16] Пул-реквест, переписывающий разбор HTML на работу с API. — (дата обращения: 12 июня 2025 г.). URL: https://github.com/IntelligeNET/HwProj-2.0.1/pull/583.

[17] Пул-реквест, позволяющий создавать курс с несколькими группами. — (дата обращения: 12 июня 2025 г.). URL: https://github.com/IntelligeNET/HwProj-2.0.1/pull/575.

[18] Пул-реквест с реализацией за осеннюю практику. — (дата обращения: 12 июня 2025 г.). URL: https://github.com/IntelligeNET/HwProj-2.0.1/pull/470.

[19] Текст работы за осенню практику. — (дата обращения: 12 июня 2025 г.). URL: https://se.math.spbu.ru/thesis/texts/Averin_Pavel_Vladislavovich_Autumn_practice_3rd_year_2024_text.pdf.