|  |  |
| --- | --- |
| RTK_LOGO.jpg | ПРАВИТЕЛЬСТВО САНКТ-ПЕТЕРБУРГА  Комитет по образованию  Санкт-Петербургское государственное бюджетное  профессиональное образовательное учреждение  «Радиотехнический колледж» |

|  |
| --- |
| К защите допущен  заместитель директора по УПР  \_\_\_\_\_\_\_\_\_\_Т.А. Бессчетнова  «\_\_\_\_» \_\_\_\_\_\_\_ 2023 г. |

**ДИПЛОМНЫЙ ПРОЕКТ**

Тема: «Автоматизированная система учёта баз практик»

|  |
| --- |
| **Специальность:** 09.02.07 «Информационные системы и программирование»  **Выполнил(а):** обучающийся группы \_\_\_\_\_\_\_ Ангелина Сергеевна Смирнова  (номер группы)  **Руководитель:** преподаватель Виталий Сергеевич Кондратьев \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Санкт-Петербург, 2023г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc137814332)

[1. ПРЕДМЕТНАЯ ОБЛАСТЬ И АНАЛОГИ 5](#_Toc137814333)

[1.1. Системный анализ предметной области. 5](#_Toc137814334)

[1.2 Обзор аналогов. 5](#_Toc137814335)

[2. ПРОЕКТИРОВАНИЕ ИС 9](#_Toc137814336)

[2.1. Описание этапов проектирования системы. 9](#_Toc137814337)

[2.2. Проектирование структуры ИС. 10](#_Toc137814338)

[2.3. Проектирование базы данных. 12](#_Toc137814339)

[2.4. Даталогическое проектирование базы данных. 19](#_Toc137814340)

[2.5. Выбор средств реализации проекта. 20](#_Toc137814341)

[3. ПОЭТАПНОЕ ОПИСАНИЕ РЕАЛИЗАЦИИ КОМПОНЕНТОВ ИС 23](#_Toc137814342)

[3.1. Подготовка. 23](#_Toc137814343)

[3.2. Создание ролей с помощью RBAC. 25](#_Toc137814344)

[3.3. Страницы авторизации и регистрации. 27](#_Toc137814345)

[3.4. Панель администрирования. 29](#_Toc137814346)

[3.5. Разработка прототипов. 30](#_Toc137814347)

[3.6. Описание страниц и их вид. 37](#_Toc137814348)

[3.7. Руководство программиста 47](#_Toc137814349)

[4. ЭКОНОМИЧЕСКАЯ ЧАСТЬ 49](#_Toc137814350)

[4.1. Определение затрат: 49](#_Toc137814351)

[4.2. Расчет себестоимости программного продукта 49](#_Toc137814352)

[ЗАКЛЮЧЕНИЕ 52](#_Toc137814353)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 54](#_Toc137814354)

[ПРИЛОЖЕНИЕ А (обязательное) Программные коды 56](#_Toc137814355)

## ВВЕДЕНИЕ

Темой дипломной работы является проектирование и разработка информационной системы (ИС) на базе Радиотехнического колледжа (РТК), которая позволяет записывать, хранить и искать производственную практику. Информационная система позволит упростить процесс поиска мест производственной практики для студентов, работодатели смогут приглашать к себе студентов и отмечать их уровень работы, а также облегчит работу колледжа и предприятий, которые предлагают свои места.

Предполагается, что студенты, перед прохождением производственной практики будут заходить на платформу и подбирать для себя нужное предприятие. А предприятия и компании, которые заинтересованы в привлечении молодых специалистов-студентов, будут размещать свои вакансии.

Производственная практика является важным периодом в ходе обучения будущих специалистов. За время прохождения производственной практики студенты учатся применять и отрабатывать свои знания и навыки, полученные во время обучения, в условиях реальной жизни. Практическая часть учебного процесса является заключительной частью учебной практики, проходящей в учебном заведении. Производственная практика, как правило, проходит на различных предприятиях в условиях реального производства.

**Актуальность**

Актуальность выбранной темы заключается в проблеме поиска места прохождения производственной практики. Зачастую в учебных заведениях возникает проблема устройства студентов на прохождение практики.

Также в связи с невероятно быстрым развитием технологий большая часть бумажной работы начала вытесняться более удобными и менее трудоемкими способами решения тех, или иных проблем, связанных с бумажной волокитой, будь это заполнение документов, написание текстов, или, как это продемонстрировано в данной работе, – формирование списков на практики.

**Цель работы**

Целью выпускной квалификационной работы является разработка автоматизированной системы учёта баз практик «Practice».

**Задачи**

* провести анализ предметной области, конкурентов и целевой аудитории;
* сформулировать требования к веб-сайту;
* обосновать выбор технических и программных средств для разработки веб-сайта;
* разработать дизайн веб-сайта;
* спроектировать и разработать веб-сайт;
* осуществить тестирование разработанной информационной системы.

**Структура дипломного проекта**

Структура дипломного проекта обусловлена актуальностью, предметом и задачами исследования. Работа включает введение, три главы, заключение, список использованных источников и приложений.

Во введении раскрыта актуальность выбранной темы, заданы предмет и объект исследования, также установлены цели и задачи, выявлена теоретическая и практическая значимость.

Первая глава посвящена определению предметной области и анализу теоретической информации по заявленной теме исследования. В первом параграфе рассматривается предметная область проекта, понятия информационной системы и производственной практики. Во втором параграфе представлены аналоги проекта, а также их описание и характеристика.

## 1. ПРЕДМЕТНАЯ ОБЛАСТЬ И АНАЛОГИ

1.1. Системный анализ предметной области.

Темой дипломной работы является разработка информационной системы (ИС) записи, хранения и поиска баз производственной практики на базе Радиотехнического Колледжа (РТК).

Предметной областью реализации данного дипломного проекта является прохождение студентами производственной практики в компаниях и на предприятиях, которые занесены в базу информационной системы.

Информационная система – это автоматизированная система, результатом функционирования которой является представление выходной информации для последующего использования, также предназначена для хранения, обработки и поиска информации.

Разработанная информационная система должна позволять:

* записывать данные в базу данных;
* хранить их;
* редактировать записи;
* удалять записи;
* иметь связь между студентом и работодателем;
* иметь связь между работодателем и студентом;
* работодатель видит резюме студента;
* студент видит данные работодателя.

1.2 Обзор аналогов.

В качестве аналогов, по данной теме может выступать большое количество проектов, поэтому были выбраны некоторые сайты, которые связаны с хранением данных.

Критерии для оценки аналогов:

1. Есть возможность авторизоваться на сайте.
2. Поиск по сайту.
3. Навигация по сайту.
4. Сайт работает без ошибок.
5. Есть возможность зарегистрироваться.
6. Соответствие целевой аудитории.
7. Удаление записей.
8. Создание публикаций.
9. Редактирование публикаций.

В качестве аналогов представлены:

1. https://stats.oecd.org/ (рисунок 1).
2. https://data.mos.ru/ (рисунок 2).
3. https://fom.ru/ (рисунок 3).



Рисунок 1 – stats.oecd.org

stats.oecd.org – сайт Организация экономического сотрудничества и развития.

Аналог позволяет осуществлять поиск по сайту, также создавать записи, редактировать их и удалять. Дизайн сайта имеет приятные цвета, за счет чего соответствует целевой аудитории.

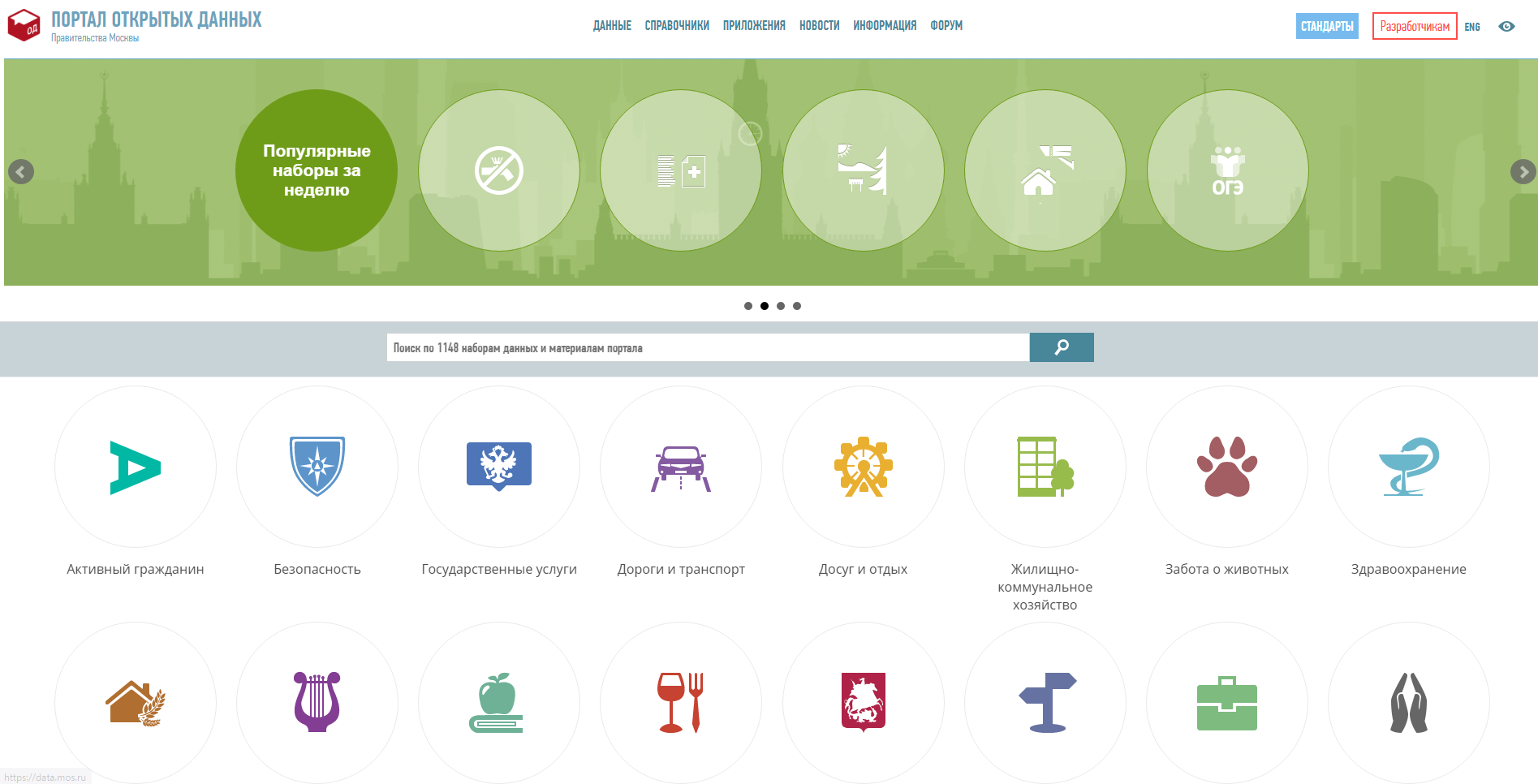


Рисунок 2 – data.mos.ru/

data.mos.ru – сайт портал открытых данных.

Позволяет пользователю осуществлять поиск по сайту, создавать публикации и редактировать их. Также сайт имеет красивый дизайн, что позволит привлекать новую аудиторию.

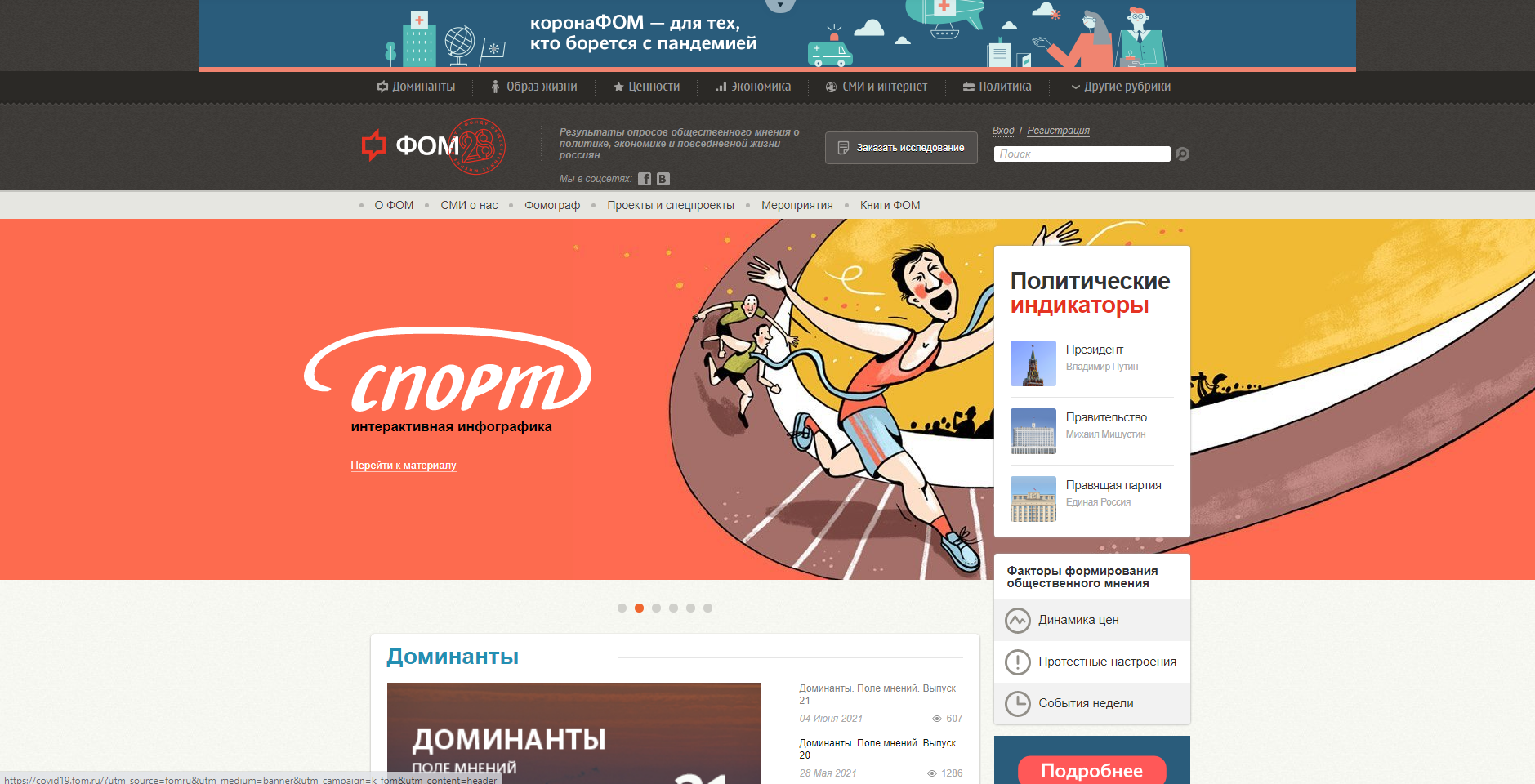


Рисунок 3 – fom.ru

Аналог, как и предыдущие, соответствует заданным критериям.

Изучив предметную область, а также разобрав перечень сервисов, предоставляющих те же возможности, что и разрабатываемый проект – получаем, что сервисы имеют как свои преимущества, так и недостатки. Рассмотрим варианты более подробно и сравним предложенные варианты между собой.

Таблица 1 – Обзор аналогов.

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | Stats.oecd.org | Data.mos.ru | Fom.ru |
| Есть возможность авторизоваться на сайте | Да | Да | Да |
| Простота в использовании, интуитивный интерфейс | Нет | Да | Да |
| Поддержка русского языка | Да | Да | Да |
| Сайт работает без ошибок | Да | Да | Да |
| Сайт работает без ошибок | Да | Да | Да |
| Редактирование публикаций | Да | Да | Да |

Исходя из проведённого исследования получаем, что актуальность разработки информационной системы подтверждена и доказана.

## 2. ПРОЕКТИРОВАНИЕ ИС

2.1. Описание этапов проектирования системы.

В системе должна быть реализована регистрация и авторизация пользователя. Эти две подсистемы служат для обеспечения парольной защиты и разграничения прав доступа в систему. С помощью регистрации устанавливаются некоторые роли пользователей, которые в дальнейшем будут разграничивать доступ к функционалу. Распределение пользователей по ролям показано на рисунке 4.

Роли

Администратор

Студент

Работодатель

Неавторизованный пользователь

Пользователь системы

Рисунок 4 – Роли в системе

В системе должны быть представлены следующие основные разделы, распределённые по ролям пользователей:

Неавторизованный пользователь:

* авторизация.

Студент:

* авторизация,
* заполнение резюме,
* просмотр работодателей,
* возможность отправить заявку на производственную практику,
* скачивание и загрузка документов,
* выход из системы.

Работодатель:

* регистрация,
* авторизация,
* заполнение резюме,
* возможность добавления в список работодателей,
* просмотр списка студентов,
* возможность пригласить студента на производственную практику,
* просмотр резюме студента,
* скачивание и загрузка документов,
* возможность автоматически заполнить данные документов,
* выход из системы.

Администратор:

* указывает период практик;
* регистрирует студентов;
* убирает отчисленных студентов;
* заносит в базу пришедших или переведённых студентов.
* выход из системы.

2.2. Проектирование структуры ИС.

В системе должна быть реализована регистрация и авторизация пользователя. Эти две подсистемы служат для обеспечения парольной защиты и разграничения прав доступа в систему. База данных хранит информацию о логинах и паролях пользователей. Подсистемы обеспечивают возможность ввода данных пользователя, проверку введенных данных, информирование о неверном вводе пароля, блокировку входа в систему при неправильном вводе данных, возможность изменения, добавления и удаления данных пользователя.

Проектируемая система должна включать в себя возможность самостоятельного выбора места производственной практики для студентов, возможность работодателям найти будущих сотрудников через приглашение студентов на производственную практику.

Для лучшего понимания структуры системы спроектирована use-case диаграмма, изображенная на рисунке 5.

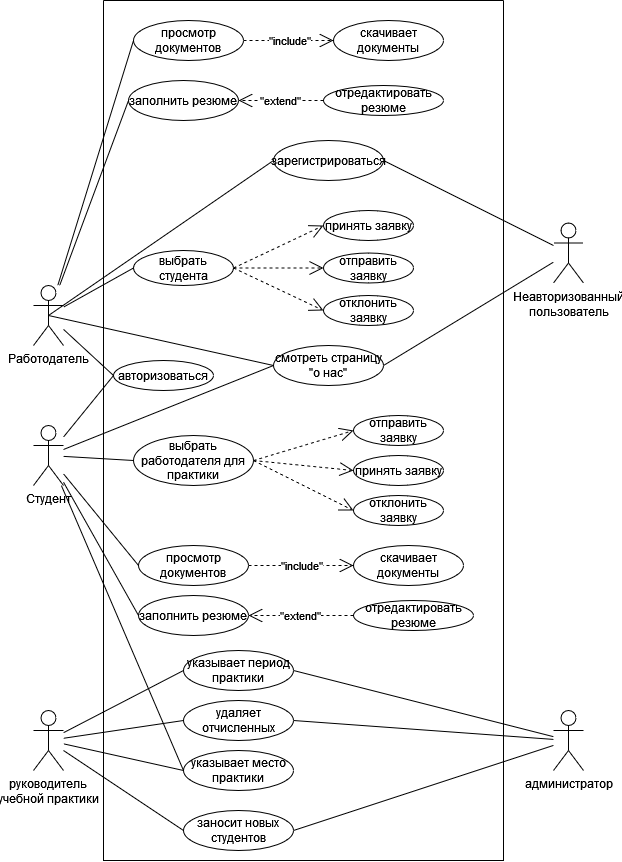


Рисунок 5 – Use-case диаграмма

Задача разрабатываемого проекта – упростить процесс поиска места производственной практики для студента. С помощью создаваемой ин-формационной системы можно будет значительно уменьшить время поис-ка практики, а также упростить процесс устройства студента. Выбрать свое место практики и устроиться сможет сам студент без третьих лиц. Все про-цессы отображены на диаграмме EPC.



Рисунок 6 – ЕРС Диаграмма

2.3. Проектирование базы данных.

Выполним концептуальное проектирование базы данных системы, необходимое для структурирования элементов при ее разработке. На этапе проектирования системы было распределено несколько ролей и выделены модули системы. Вследствие этого, база данных будет включать в себя упомянутые выше сущности, их связи и атрибуты.

Для начала в базе данных была создана таблица пользователей – user. В таблице 2 указаны поля таблицы user.

Таблица 2 – Таблица user.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя столбца** | **Тип** | **Атрибуты** | **По умолчанию** | **Ключевое поле** |
| id | int(10) | UNSIGNED |  | auto\_increment |
| name | varchar(255) |  |  |  |
| surname | varchar(255) |  |  |  |
| patronymic | varchar(255) |  | NULL |  |
| login | varchar(255) |  |  |  |
| email | varchar(255) |  |  |  |
| password | varchar(255) |  |  |  |
| auth\_key | varchar(255) |  |  |  |
| course | int(10) |  | NULL |  |
| Specialization\_id | int(10) |  | NULL |  |

В разрабатываемой системе используется RBAC (контроль доступа на основе ролей), благодаря этому механизму автоматически создаются таблицы в базе данных, которые отвечают за разграничение ролей.

Инструмент RBAC использует пять таблиц для хранения данных:

1. auth\_rule: таблица для хранения правил, данные представлены в таблице 3.
2. auth\_item: таблица для хранения авторизационных элементов, данные представлены в таблице 4.
3. auth\_item\_child: таблица для хранения иерархии элементов, данные представлены в таблице 5.
4. auth\_assignment: таблица для хранения назначений элементов авторизации, данные представлены в таблице 6.
5. migration: таблица для хранения миграций, данные представлены в таблице 7.

Таблица 3 – Таблица auth\_rule.

|  |  |  |
| --- | --- | --- |
| **Имя столбца** | **Тип** | **По умолчанию** |
| name | varchar(64) |  |
| data | blob | NULL |
| created\_at | int(11) | NULL |
| updated\_at | int(11) | NULL |

Таблица 4 – Таблица auth\_item.

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя столбца** | **Тип** | **По умолчанию** | **Связи** |
| name | varchar(64) |  |  |
| title | varchar(255) | NULL |  |
| type | smallint(6) |  |  |
| description | text | NULL |  |
| rule\_name | varchar(64) | NULL | Таблица auth\_rule, атрибут name |
| data | blob | NULL |  |
| created\_at | int(11) | NULL |  |
| updated\_at | int(11) | NULL |  |

Таблица 5 – Таблица auth\_item\_child.

|  |  |  |
| --- | --- | --- |
| **Имя столбца** | **Тип** | **Связи** |
| parent | varchar(64) | Таблица auth\_item, атрибут name |
| child | varchar(64) | Таблица auth\_item, атрибут name |

Таблица 6 – Таблица auth\_assignment.

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя столбца** | **Тип** | **По умолчанию** | **Связи** |
| item\_name | varchar(64) |  | Таблица auth\_item, атрибут name |
| user\_id | varchar(64) |  |  |
| created\_at | int(11) | NULL |  |

Таблица 7 – Таблица migration.

|  |  |  |
| --- | --- | --- |
| **Имя столбца** | **Тип** | **По умолчанию** |
| version | varchar(180) |  |
| apply\_time | int(11) | NULL |

Также были спроектированы основные таблицы для работоспособности проекта.

Таблица 8 – Таблица documents.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя столбца** | **Тип данных** | **Null** | **Дополнительно** | **Комментарии** |
| id | |  | | --- | | int | | Нет | AUTO\_INCREMENT | Первичный |
| practice\_diary | varchar(255) | Да |  | Дневник по практике |
| characteristic | varchar(255) | Да |  | Характеристика |
| practical\_task | varchar(255) | Да |  | Приложение к договору о практике |
| certification\_sheet | varchar(255) | Да |  | Аттестационный лист |
| contract | varchar(255) | Да |  | Договор о практической подготовке обучающихся |

Таблица 9 – Таблица student\_lists.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя столбца** | **Тип данных** | **Null** | **Дополнительно** | **Комментарии** |
| id | int | Нет | AUTO\_INCREMENT | Комментарии |
| student\_id | int | Нет |  | Связь с таблицей студент |
| specialization\_  id | int | Нет |  | Связь с таблицей специальности |
| practice\_date\_  from | timestamp | Нет |  | Дата начало практики |
| practice\_date\_  to | timestamp | Нет |  | Дата конец практики |

Таблица 10 – Таблица employer\_lists.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя столбца** | **Тип данных** | **Null** | **Дополнительно** | **Комментарии** |
| id | int | Нет | AUTO\_INCREMENT | Комментарии |
| employer\_id | int | Нет |  | Связь с таблицей работодатель |
| information\_id | int | Нет |  | Связь с таблицей информация |

Таблица 11 – Таблица information.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя столбца** | **Тип данных** | **Null** | **Дополнительно** | **Комментарии** |
| id | int | Нет | AUTO\_INCREMENT | Первичный |
| name | varchar(255) | Нет |  | Имя |
| surname | varchar(255) | Нет |  | Фамилия |
| patronymic | varchar(255) | Нет |  | Отчество |
| address | varchar(255) | Нет |  | Адрес |
| organization\_  name\_id | int | Нет |  | Связь с таблицей наименование организации |
| email | varchar(255) | Нет |  | Почта |
| phone | varchar(255) | Нет |  | Телефон |

Таблица 12 – Таблица student\_practice.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя столбца** | **Тип данных** | **Null** | **Дополнительно** | **Комментарии** |
| id | int | Нет | AUTO\_INCREMENT | Первичный |
| student\_id | int | Нет |  | Связь с таблицей пользователей |
| practice\_group\_id | int | Нет |  | Связь с таблицей практик групп |
| organization\_id | int | Да |  | Связь с таблицей организаций |
| place\_title | text | Да |  | Наименование места |

Таблица status предназначена для определения статусов задач и ответов. В таблице 13 указаны поля таблицы status.

Таблица 13 – Таблица status.

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя столбца** | **Тип** | **Атрибуты** | **Ключевое поле** |
| id | int(10) | UNSIGNED | auto\_increment |
| title | varchar(255) |  |  |

Таблица 14 – Таблица goup.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя столбца** | **Тип данных** | **Null** | **Дополнительно** | **Комментарии** |
| id | int | Нет | AUTO\_INCREMENT | Первичный |
| title | varchar(255) | Нет |  | Наименование |
| specialization\_id | int | Нет |  | Связь с таблицей специальностей |

Таблица 15 – Таблица student\_group.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя столбца** | **Тип данных** | **Null** | **Дополнительно** | **Комментарии** |
| id | int | Нет | AUTO\_INCREMENT | Первичный |
| student\_id | int | Нет |  | Связь с таблицей пользователей |
| group\_id | int | Нет |  | Связь с таблицей группы |

Таблица 16 – Таблица view\_practice.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **-Имя столбца** | **Тип данных** | **Null** | **Дополнительно** | **Комментарии** |
| id | int | Нет | AUTO\_INCREMENT | Первичный |
| title | varchar(255) | Нет |  | Наименование |

Таблица 17 – Таблица practice\_group.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя столбца** | **Тип данных** | **Null** | **Дополнительно** | **Комментарии** |
| id | int | Нет | AUTO\_INCREMENT | Первичный |
| view\_practice\_id | int | Нет |  | Связь с таблицей видами практик |
| group\_id | int | Нет |  | Связь с таблицей группы |
| begin\_date | timestamp | Да |  | Дата начала |
| end\_date | timestamp | Да |  | Дата конца |

Для наглядности, в сервисе [Draw.io](https://app.diagrams.net/) мной была создана диаграмма «Сущность-связь» спроектированной базы данных, представленная на рисунке 7.

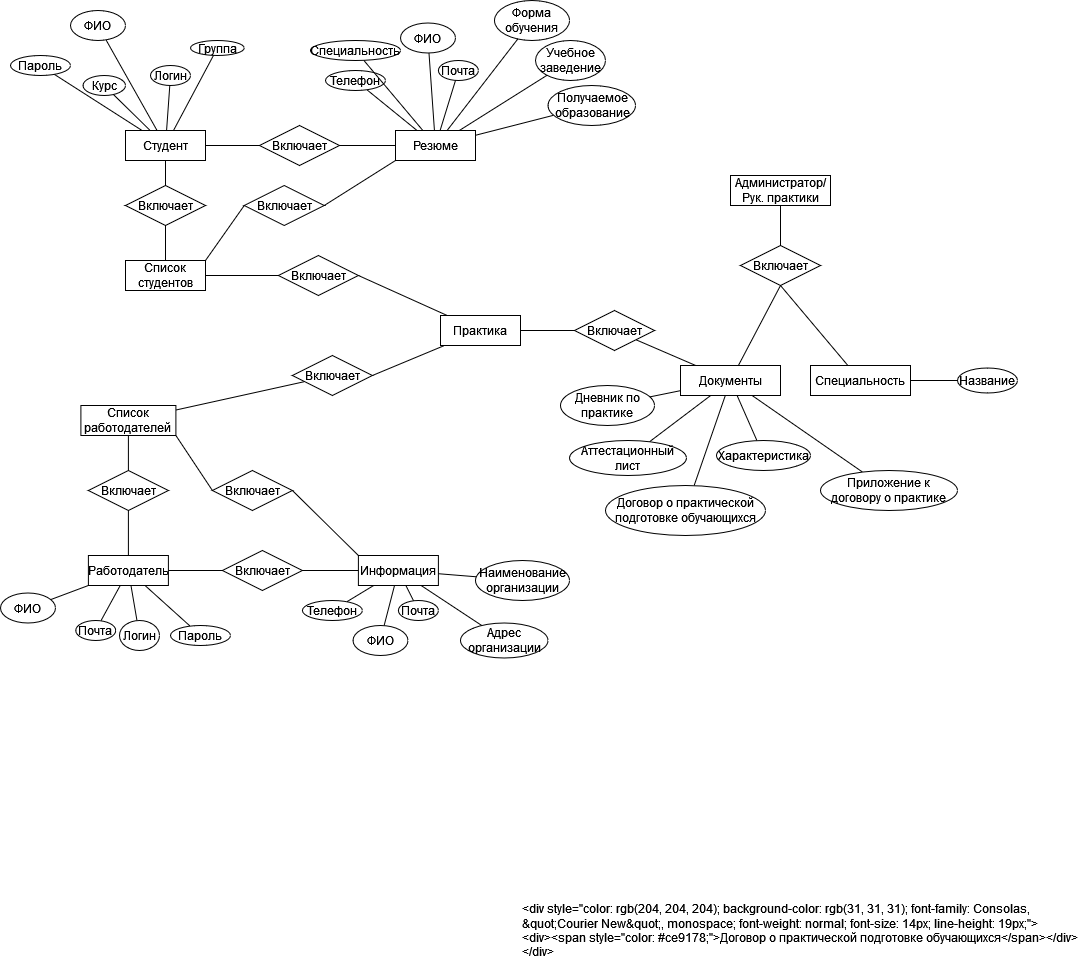


Рисунок 7 - Диаграмма «Сущность-связь»

Графически сущности на диаграммах «сущность-связь» представляются в виде прямоугольников, изображение атрибутов сущности представляются в виде овальных блоков, а для графического представления связи используется ромб. Показатель кардинальности связи (характеристика однозначности) обозначает степень взаимосвязи сущностей и описывает количество возможных связей для каждой из сущностей-участниц:

* один-к-одному (1:1),
* один-ко-многим (1:N),
* многие-ко-многим (N:M).

Была создана полная атрибутивная модель, полностью удовлетворяющая процессу нормализации данных. В результате проведения нормализации была создана структура данных, при которой информация о каждом факте хранится только в одном месте. Процесс нормализации сводится к последовательному приведению структуры данных к нормальным формам – формализованным требованиям к организации данных.

2.4. Даталогическое проектирование базы данных.

Разработка базы данных является непростой и трудоемкой задачей, так как именно она считается одним из самых важных элементов для информационной системы. При разработке учитывался проведенный ранее предпроектный анализ.

Разработанная база данных изображена на рисунке 8.

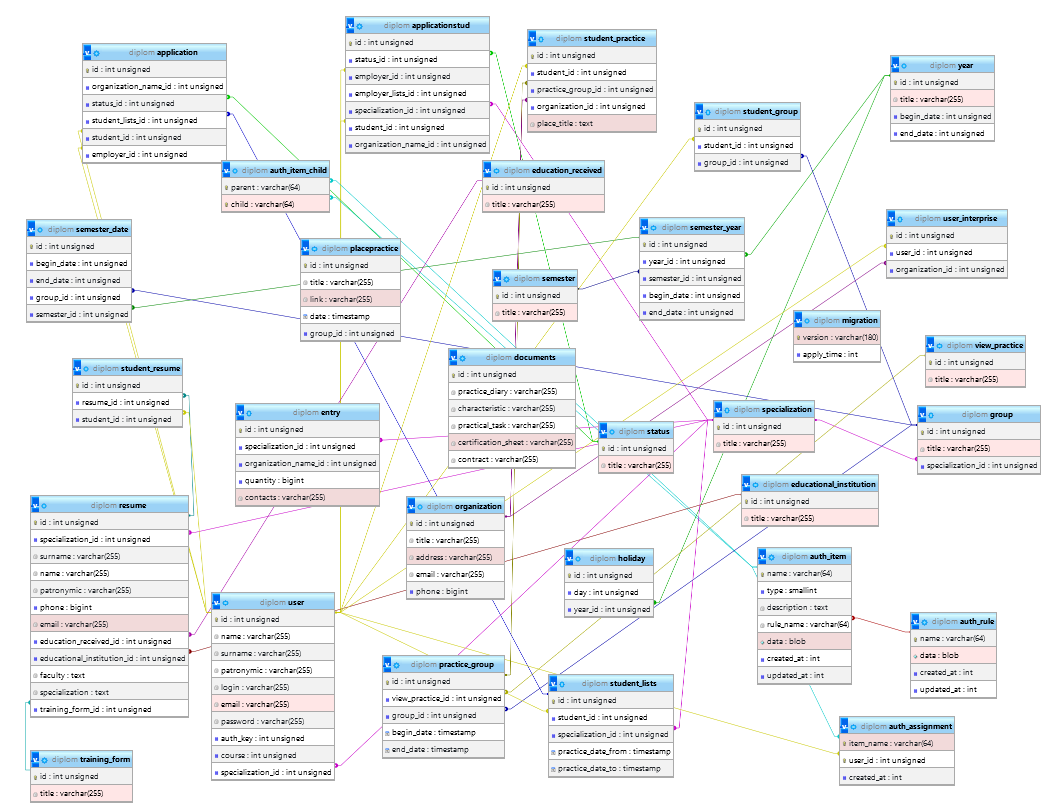


Рисунок 8 – База данных

База данных выполняет следующие функции:

* хранение информации о товарах,
* хранение архива заказов по всем клиентам,
* роли данного сайта,
* оформление заказов,
* в базе можно отслеживать статус заказа.

2.5. Выбор средств реализации проекта.

Тщательно подойдя к выбору технологий для разработки моего проекта, составлено краткое описание преимуществ выбранных платформ.

Язык программирования Php:

* простота использования, простой синтаксис;
* легкость изучения (доступные учебные пособия) и читабельность кода;
* гибкость, т.е. способность смешиваться с другими языками;
* работа с фреймворком Yii2;
* универсальность (подходит для различных проектов).

Фреймворк Yii2:

* простой и быстрый (создание сайта из компонентов);
* функциональность (административная панель, формы, миграции и др.);
* безопасность;
* работа с крупными проектами и большой загруженностью;
* существует два заготовленных шаблона проекта для разработчиков с разным уровнем подготовки;
* имеет веб-инструмент для генерации кода, что обеспечивает быстроту его написания;
* имеет замечательную документацию, гайды по старту.

База данных MySQL:

* высокая скорость и производительность,
* простота использования,
* безопасность и простота установки,
* бесплатное обслуживание и открытый доступ.

HTML, CSS и JS фреймворк Bootstrap:

* высокая скорость создания сайта;
* корректная работа сайта во всех поддерживаемых браузерах и ос;
* готовые компоненты и шаблоны;
* элементарные навыки верстки;
* однородность дизайна;
* легкость изучения (огромное количество обучающих материалов).

Среда разработки Visual Studio Code:

* помощь при написании кода (функции автодополнения и анализа кода, подсветка и исправление ошибок);
* встроенные инструменты (инструмент запуска тестов, терминал, инструменты для работы с базами данных);
* поддержка различных веб-фреймворков.

Локальный сервер OpenServer:

* не требует установки на компьютер — его можно запустить с флешки или внешнего жёсткого диска;
* быстро запускается и останавливается;
* сервер автоматически начинает работу при запуске программы;
* можно выбрать режим управления доменами;
* возможна работа через командную строку;
* можно переключать модули http, mysql, php;
* достаточно одного клика для доступа к доменам;
* интерфейс поддерживает несколько языков.

## 3. ПОЭТАПНОЕ ОПИСАНИЕ РЕАЛИЗАЦИИ КОМПОНЕНТОВ ИС

Сами страницы подразделены на логические и тематические блоки в зависимости от их наполнения. Для быстрой вёрстки HTML-кода веб-приложения был выбран HTML, CSS и JS фреймворк Bootstrap. Благодаря наличию у него большого количества готовых шаблонов, можно создавать сайт с красивым, приятным, профессиональным дизайном за достаточно короткое время.

3.1. Подготовка.

После развертывания проекта на сервере, в файле config/db.php указываются данные для подключения к базе данных – ее название, логин и пароль для авторизации.

В config/web.php необходимо указать различные параметры сайта:

* name – название, указываемое в шапке сайта;
* language – язык сайта;
* cookieValidationKey – ключ для проверки дополнительная безопасность проверки cookie, которая защищает файлы cookie от изменения на стороне клиента;
* baseUrl – базовый путь.

Также раскомментируется блок urlManager в конце файла для активации ЧПУ.

После успешного подключения БД создаются модели таблиц (рисунок 9).

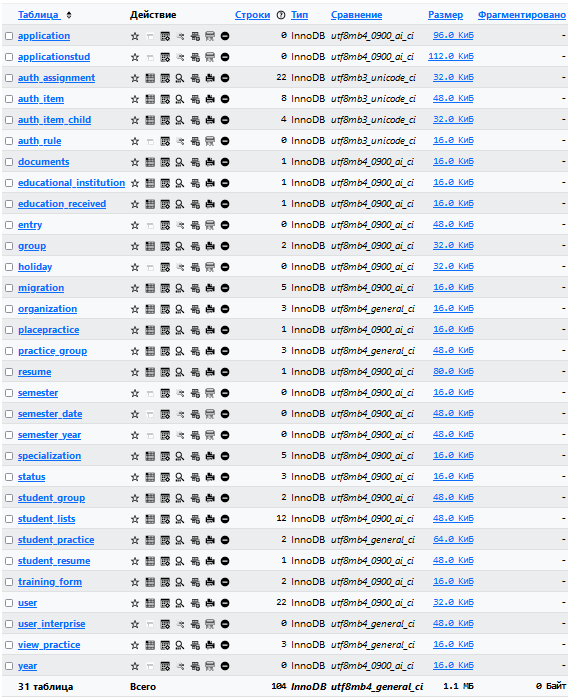


Рисунок 9 – Модели таблиц

Далее требуется применить шаблон к системе. Сначала css и js файлы перемещаются в отдельную папку в директории web. Прописывается вывод контента на шаблон через вывод переменной $content.

Для корректного импорта стилей и скриптов создаётся отдельный AssetBundle с названием AppAsset, как указано на рисунке 10. В нем указываются все стили и скрипты из шаблона.

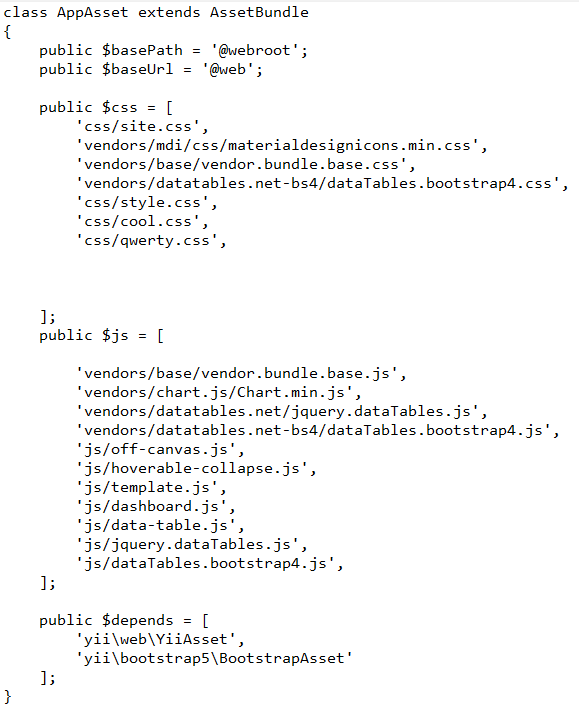


Рисунок 10 – Класс AppAsset

3.2. Создание ролей с помощью RBAC.

RBAC (Role Based Access Control) в php-фреймворке Yii2 переводится как система доступа на основе ролей.

Чтобы подключить систему, необходимо настроить компонент приложения authManager как в файле config/web.php, так и в конфигурации консольного приложения config/console.php (рисунок 11).



Рисунок 11 – Настройка DbManager

Прежде чем начать использовать этот менеджер, нужно создать необходимые таблицы в базе данных. Чтобы сделать это, в консоли прописывается следующая миграция:

yii migrate --migrationPath=@yii/rbac/migrations.

Теперь authManager может быть доступен через \Yii::$app->authManager.

RBAC позволяет создать роли автоматически с помощью консольного скрипта. Этот скрипт прописывается в файле RbacController.php. На стадии проектирования были определены основные роли пользователей. Таким образом, получились разграничения ролей, указанные на рисунке 12.

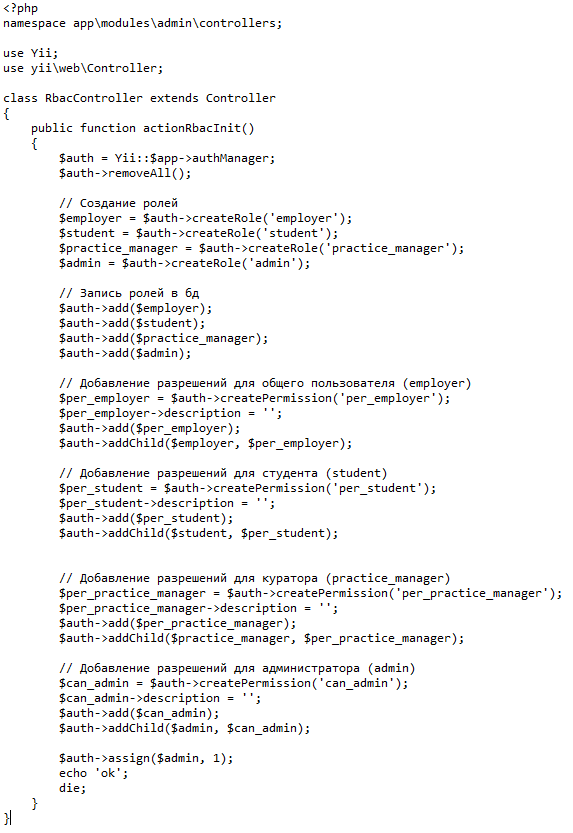


Рисунок 12 – RbacController

После выполнения команды yii rbac/init, в созданные ранее таблицы базы данных добавятся заполненные данные.

3.3. Страницы авторизации и регистрации.

Для начала, в классах RegisterForm и LoginForm прописываются свойства и правила валидации полей форм в соответствии с их назначением.

Создается и сохраняется модель пользователя с указанными данными в методе registerUser, код которого представлен на рисунке 13.

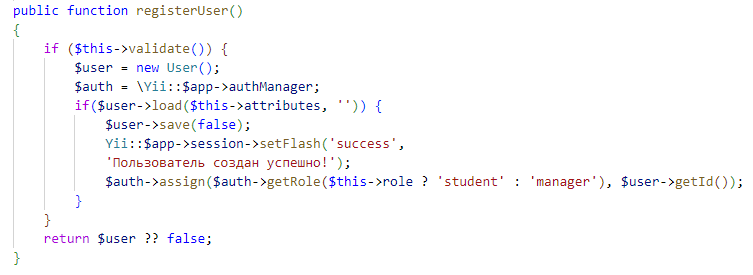


Рисунок 13 – Метод registerUser

Благодаря authManager в методе реализовано присвоение пользователю одной из ролей.

Нажав на ссылку регистрации или входа, пользователю открывается соответствующая страница, на которой предлагается заполнить поля, необходимые для того, чтобы зарегистрироваться в системе или же войти в уже существующий аккаунт. Этот механизм реализован с помошью методов actionLogin и actionRegister, которые принадлежат контроллеру SiteController.

В методе actionRegister (рисунок 14) происходит рендеринг страницы регистрации и вызов метода registerUser с последующей авторизацией и переадресацией на главную страницу.



Рисунок 14 – Метод actionRegister

Класс LoginForm дорабатывается в соответствии с заданием – свойство username переименовывается в login, добавляется вывод flash-сообщения об ошибке в случае, если введенные данные неверны (рисунок 15).

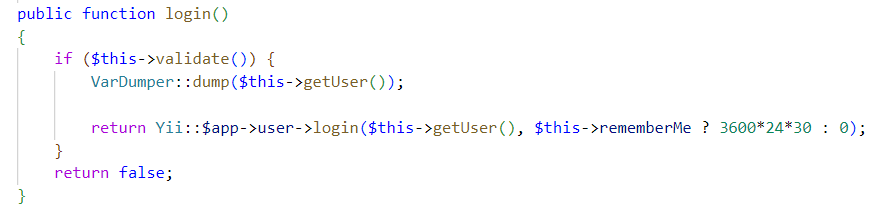


Рисунок 15 – Метод login

3.4. Панель администрирования.

В Gii создаётся модуль admin. Далее генерируются CRUD для всех моделей, кроме user и role. Все CRUD должны находиться внутри модуля. Для корректной работы необходимо включить Pjax при генерации. Пример заполненной формы показан на рисунке 16.

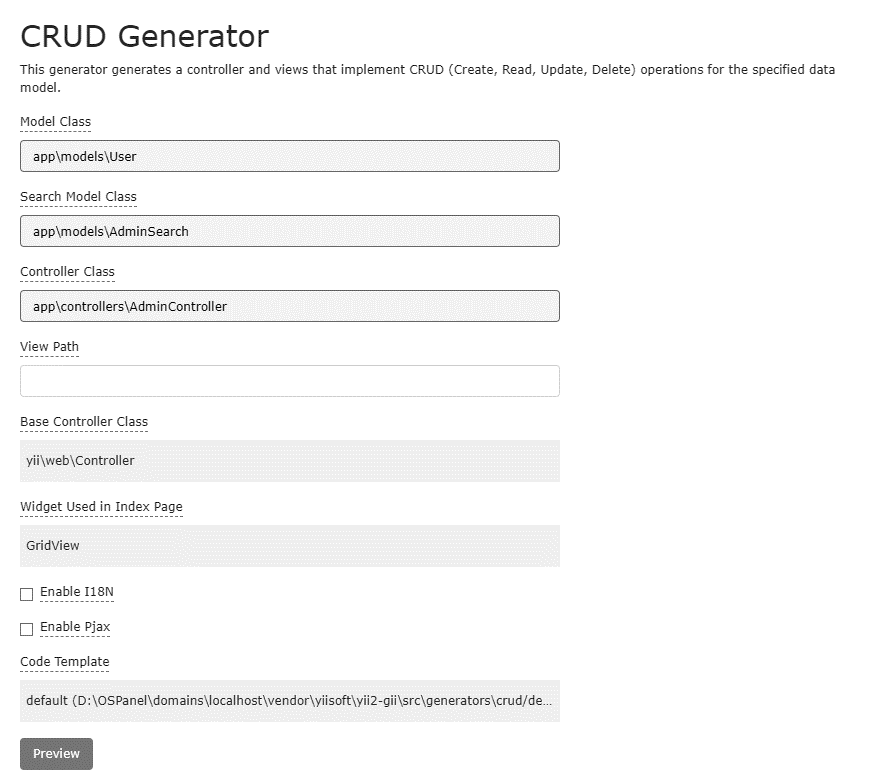


Рисунок 16 – Генерация CRUD

Для каждого CRUD настраивается вывод данных – добавляются выпадающие списки, настраивается фильтрация и сортировка. Для ввода дат используется библиотека.

3.5. Разработка прототипов.

Создание прототипа является важной частью проекта по разработке веб-сайта. Когда клиент получает четкое визуальное представление о сайте в самом начале, это дает возможность для дизайна еще на этапе концепции и практически устранить риск недопонимания. Это экономит время, когда команда концентрируется на создании фактического дизайна и программирования.

Прототип главной страницы.

На странице «О нас» отображается информация о данном сайте, чтобы пользователю было понятно как устроен сайт, и какие цели несёт в себе.

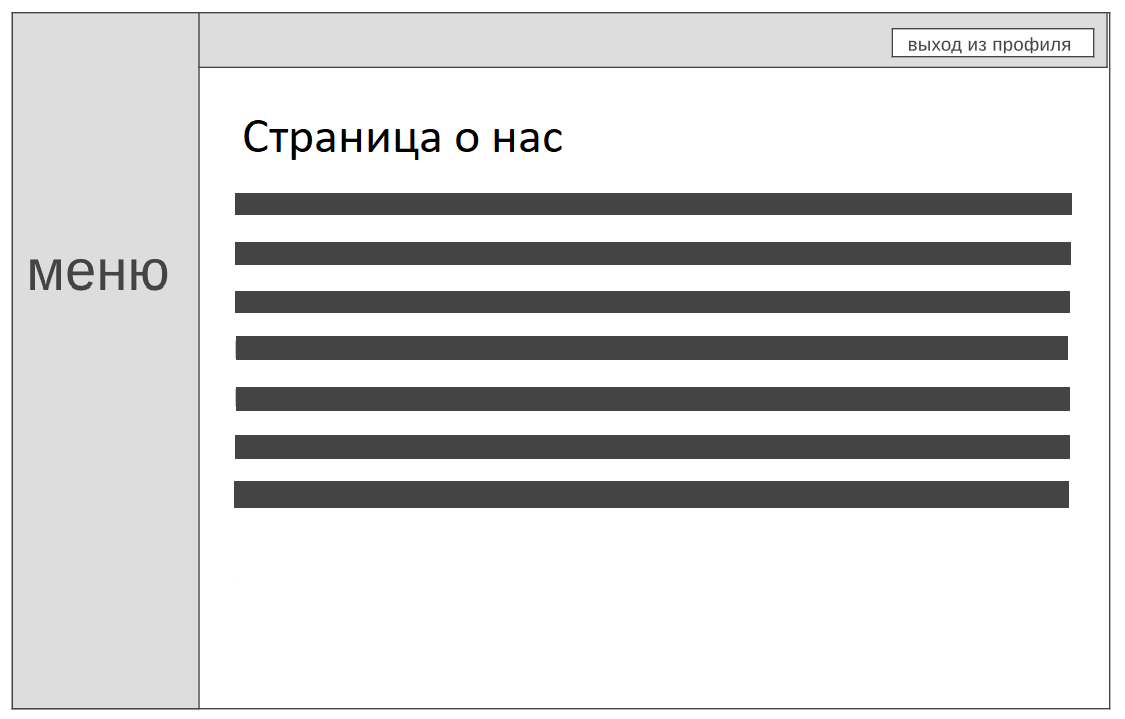


Рисунок 17 – Прототип главной страницы

Прототипы страниц студента.

На странице студента отображаются поля для резюме. Резюме нужно, чтобы работодатель видел интересующие его данные.

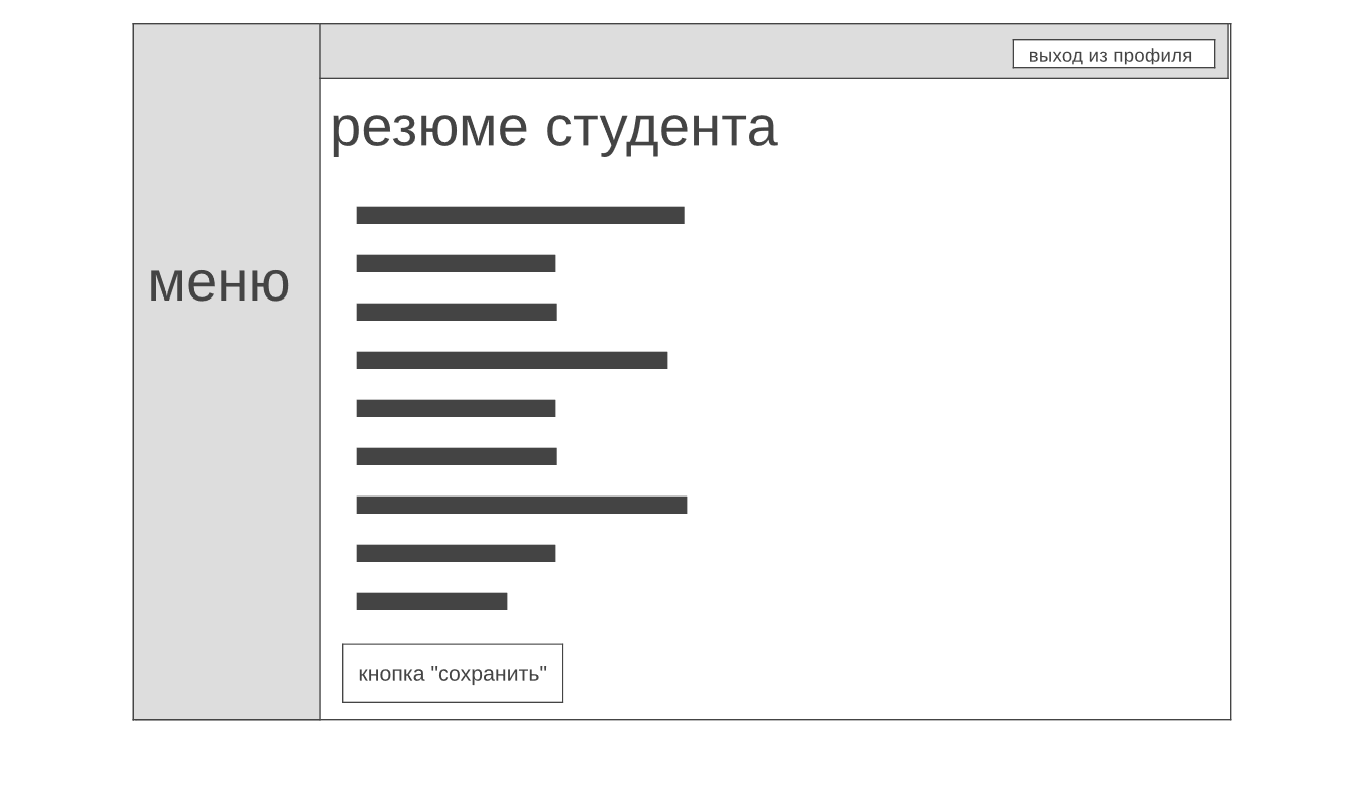


Рисунок 18 – Прототип резюме студента

После заполнения резюме профиль студента выглядит следующим образом.

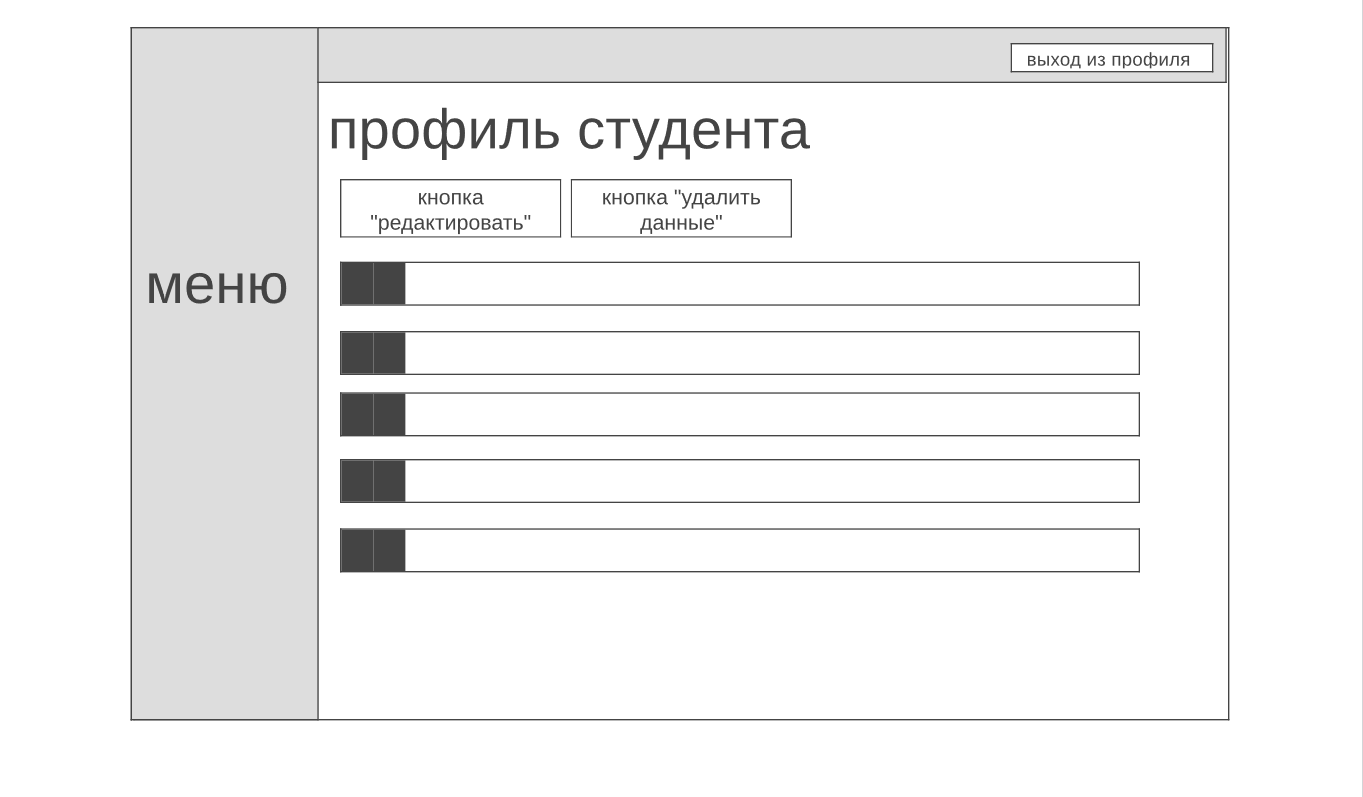


Рисунок 19 – Прототип страницы студента

Прототипы страниц работодателя.

На странице работодателя отображаются поля для резюме. Резюме нужно, чтобы студенты видели интересующие их данные по месту прохождения практики.

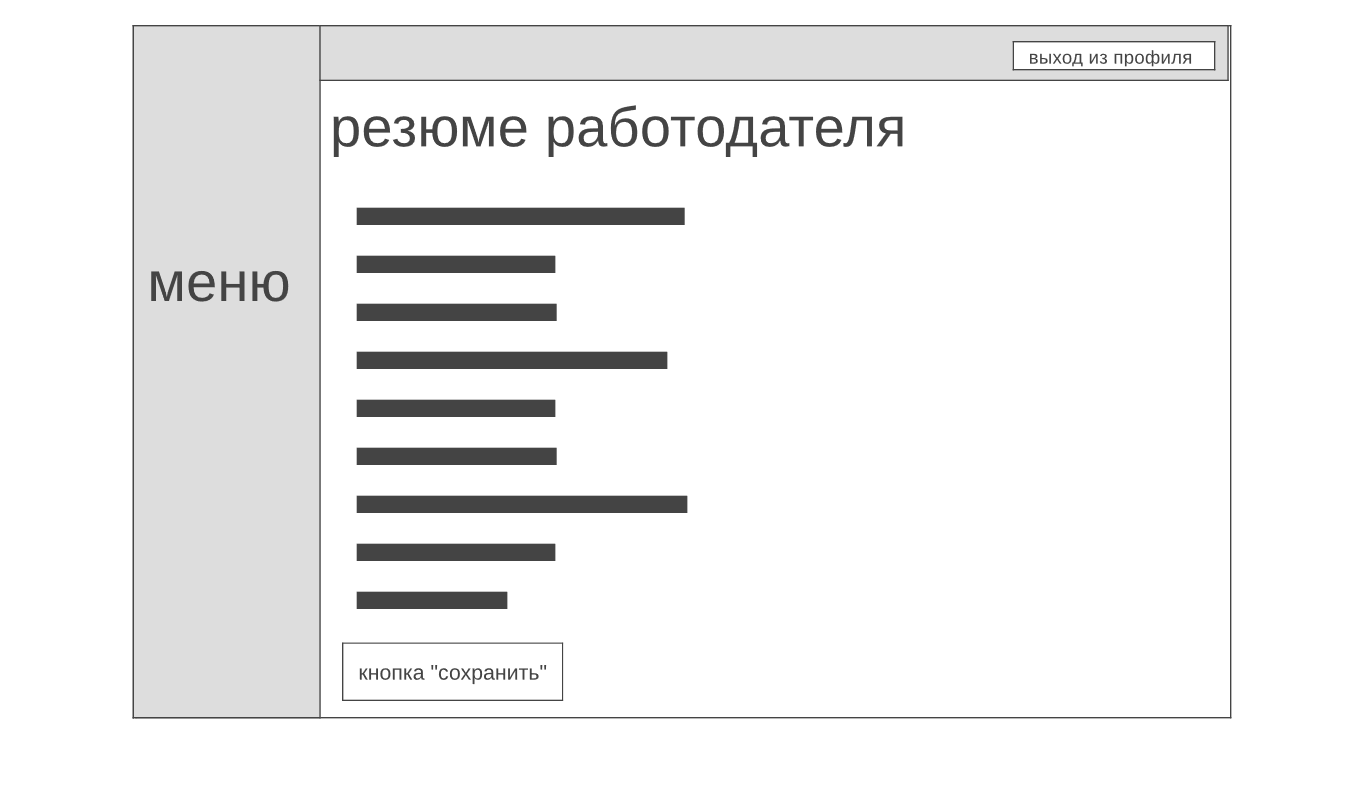


Рисунок 20 – Прототип резюме работодателя

После заполнения резюме профиль студента выглядит следующим образом.

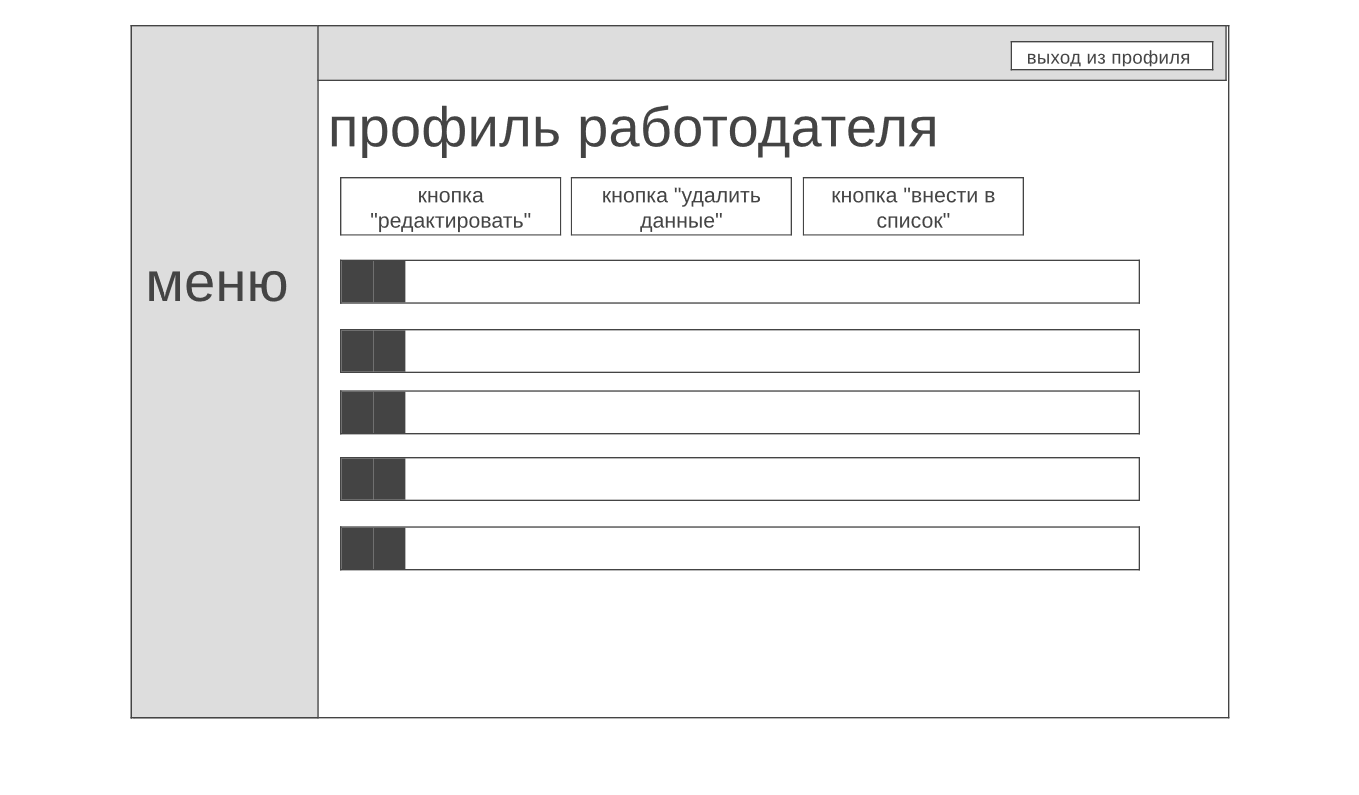


Рисунок 21 – Прототип страницы работодателя

На данной страницы работодатели могут просматривать списки студентов, смотреть интересующие их данные, а также отправлять им приглашение на производственную практику.

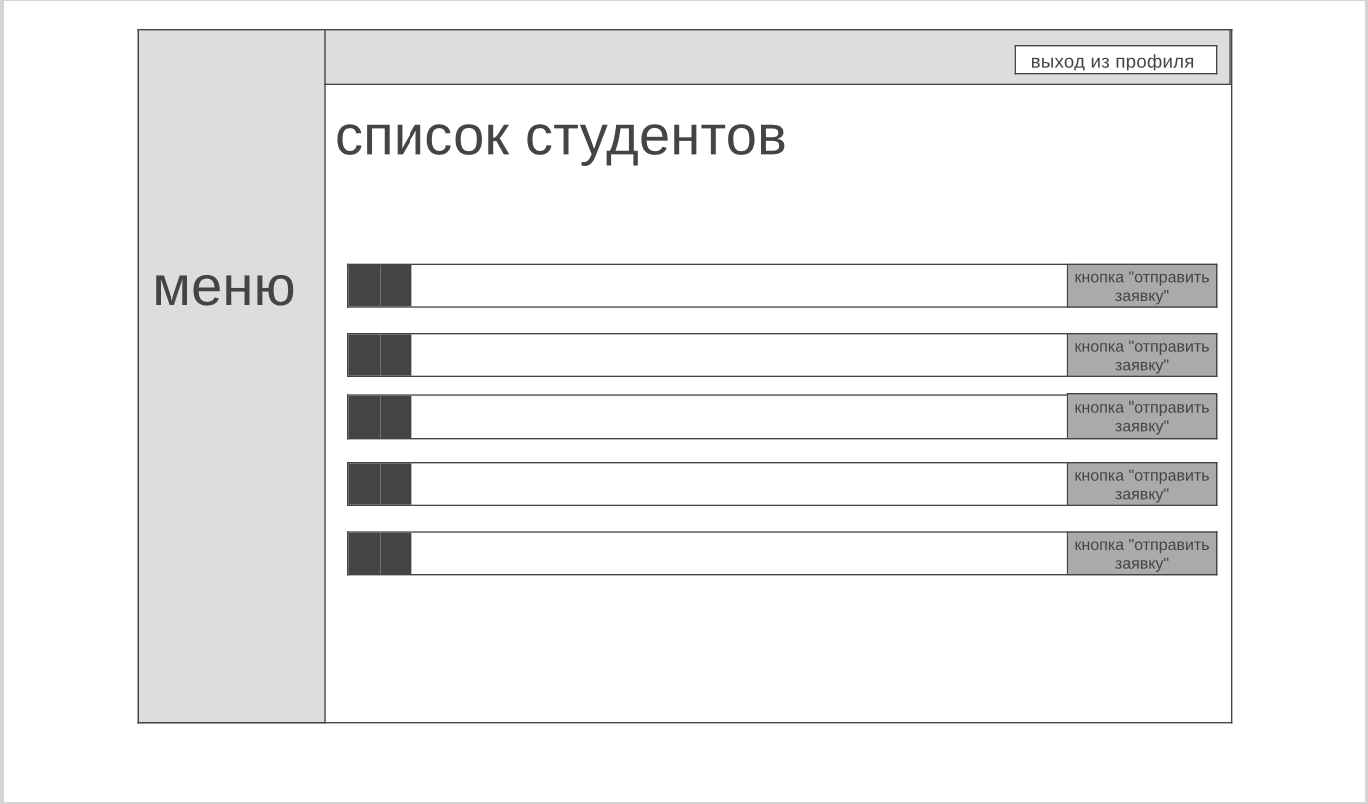


Рисунок 22 – Прототип страницы списка студентов

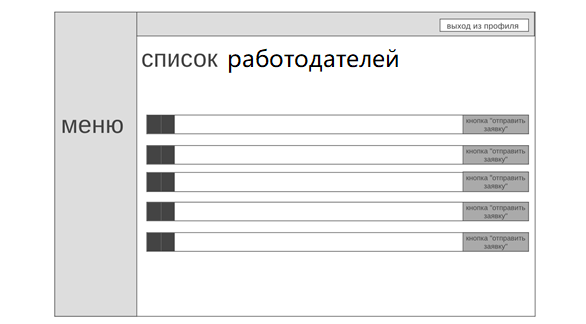


Рисунок 23 – Прототип страницы списка работодателей

Прототипы страниц администратора.

Для администратора доступны такие страницы, как «специальность» и «контроль доступа к практике» (Рисунок 24 и Рисунок 25)

На странице «специальность» администратор может создавать, редактировать, удалять и просматривать текущие специальности в учебном заведении.

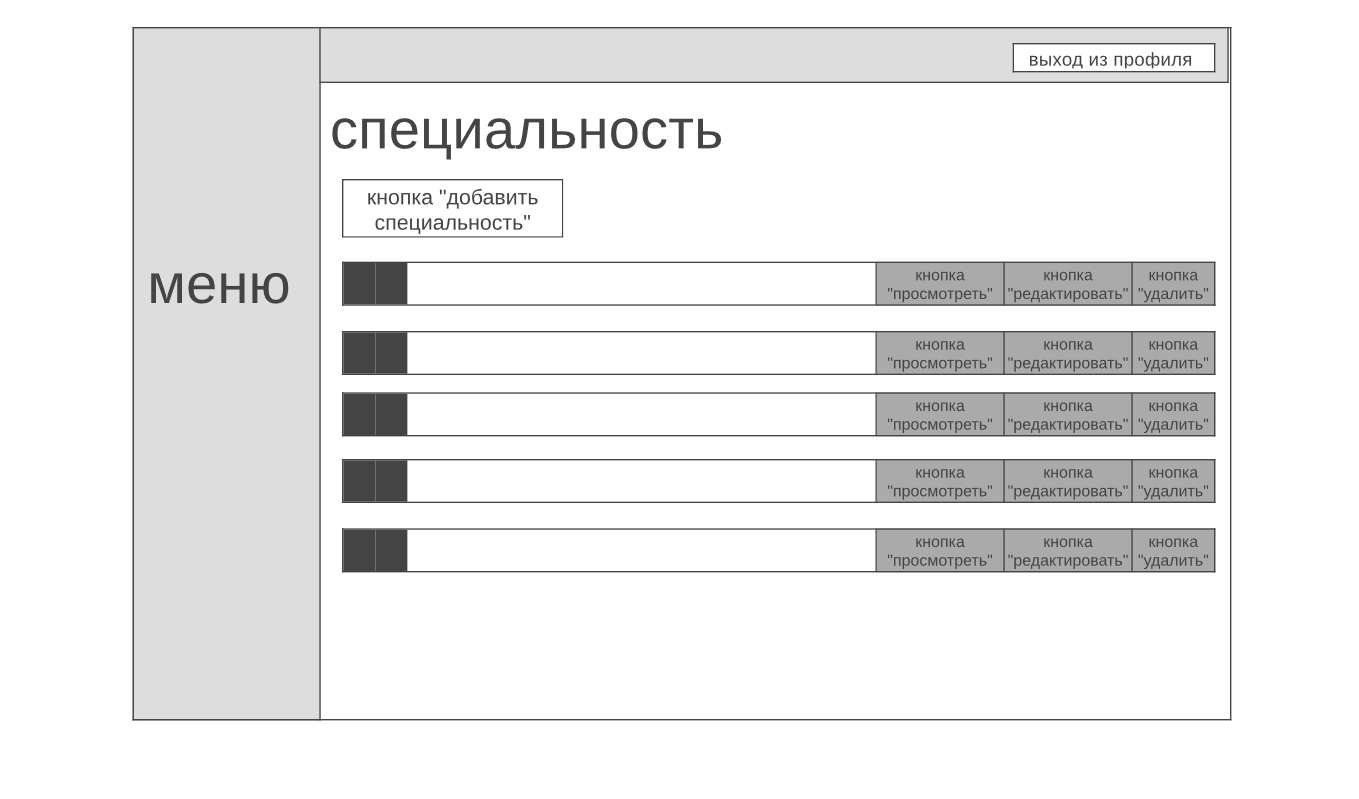


Рисунок 24 – Прототип страницы специальностей

Страница контроля доступа нужна, чтобы студент имеющий учебные задолженности понимал, что его не допускают к практике.

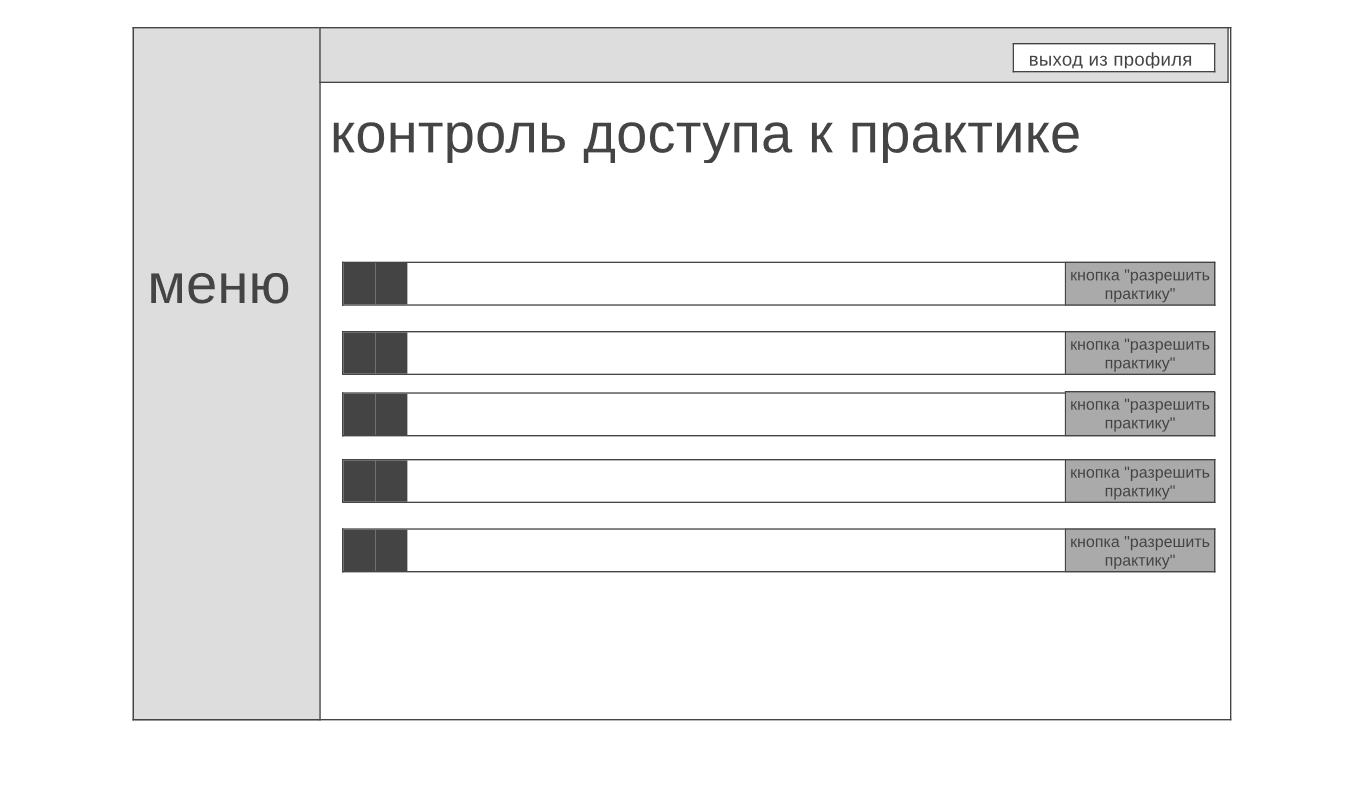


Рисунок 25 – Прототип страницы контроля доступа к практике

Прототип страницы авторизации.

На странице авторизации представлена форма с двумя полями. Т.к регистрироваться пользователи будут напрямую через базу данных, больше сведений не нужно.

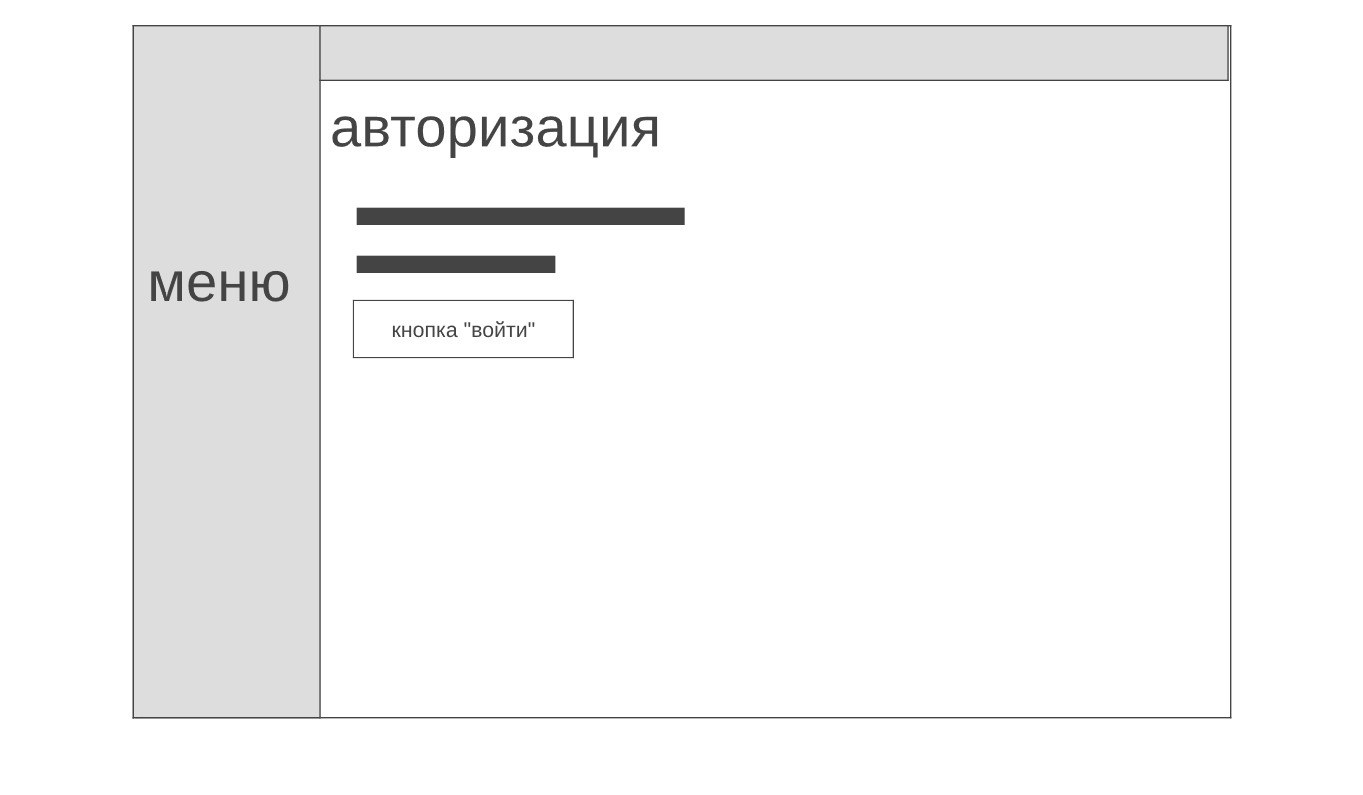
****

Рисунок 26 – Прототип страницы авторизации

Регистрация доступна только работодателя, т.к. руководитель уже имеет личный аккаунт, а студент регистрирует админ.

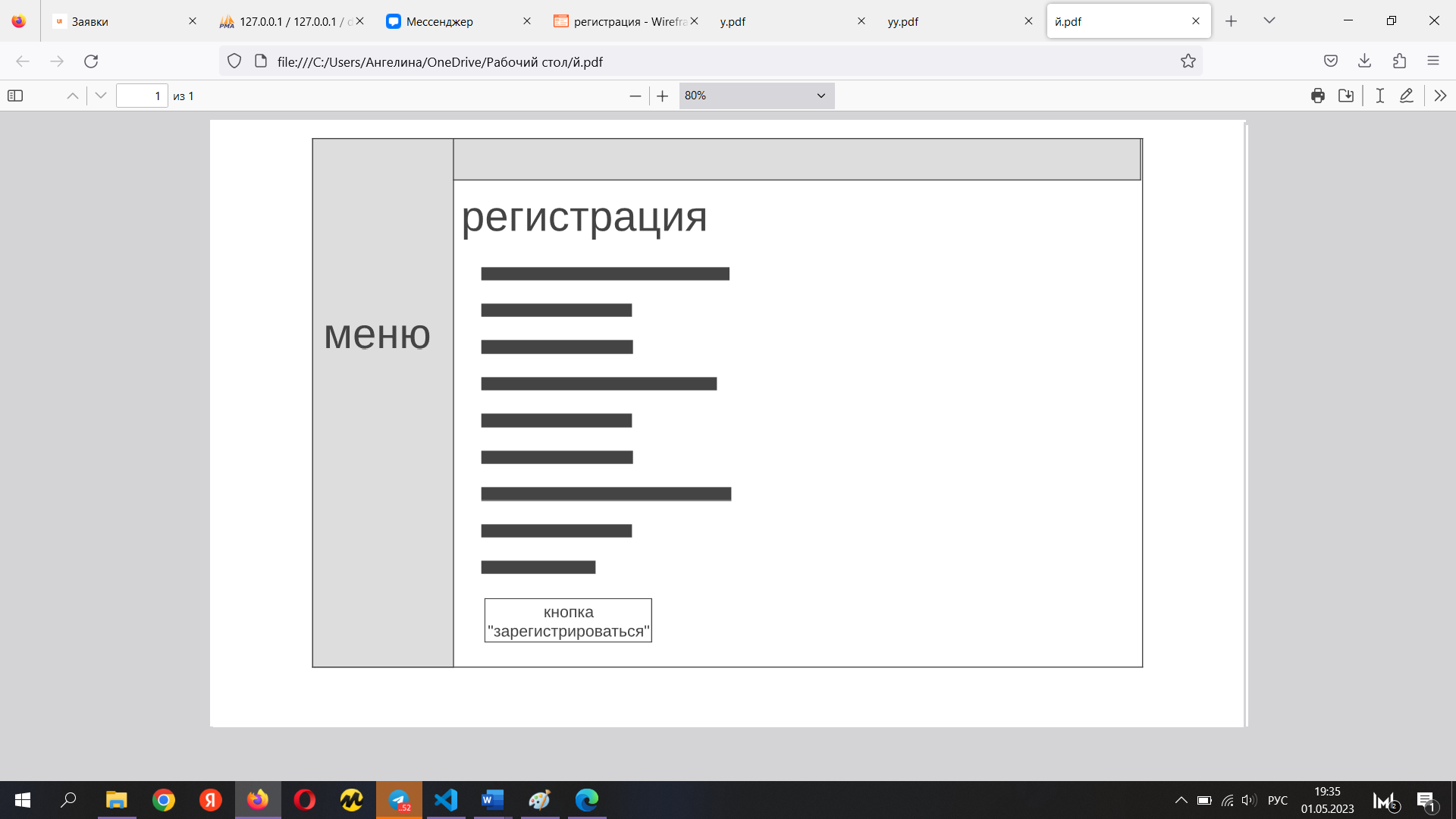


Рисунок 27 – Прототип страницы регистрации

Таким образом был спроектирован прототип информационной системы и следующим шагом стала верстка по данным шаблонам.

3.6. Описание страниц и их вид.

В ходе реализации была использована вся информация и наработки, которые использовались в предыдущих главах.

Первая страница с которой сталкивается пользователь – это страница «О нас». На данной странице представлена краткая информация о сайте и контакты для связи. Далее неавторизованный пользователь – это работодатель, он может зарегистрировать и соответственно авторизоваться (Рисунок 28-29). После заполнения резюме (Рисунок 30), работодатель может занести себя в «Список работодателей» (Рисунок 31), чтобы студенты могли отправлять свои заявки к ним на производственную практику. Также и сам работник может отправлять заявки студентам, с приглашение на практику, зайдя на страницу «Списки студентов» (Рисунок 32). Работодатель может просматривать резюме студента, скачивать документы по практике (Рисунок 33), а также для удобства может сразу заполнять форму с оценками до распечатки документов (Рисунок 34).

Что касается функционала студента, то его регистрирует администратор, после ученик может заполнить своё резюме, указать место производственной практики (Рисунок 35), на данной странице идёт ссылка на google-документ с таблицей данных по месту практики, распечатать документы, и также пройти на страницу «Списки работодателей», выбрать понравившийся «ПМК» и отправить заявку.

Функционал администратора и руководителя практики идентичен. Они могут добавлять новые специальности при необходимости (Рисунок 36), просматривать место практики студентов, и заполнять форму по истечению срока. И самое главное, делают проверку студентов на странице «Контроль доступа к практике» (Рисунок 37). На данной странице нужно выбрать студента, который допущен до практики и указать его время практики, данные студента переносятся на страницу «Список студентов». Также админ может добавить группу на практику, указав период практики, группу и вид практики (Рисунок 38). После чего на странице «Группы на практику» (Рисунок 39) отображаются группы с определённым видом практики, с данной страницы, админ может отправить всю группу на практику с указанием места прохождения. На странице «Студенты группы на практику» (Рисунок 40) отображаются студенты выбранной группы и можно индивидуально изменить место практики, выбрав студента и указав новое место в модальном окне (Рисунок 41).

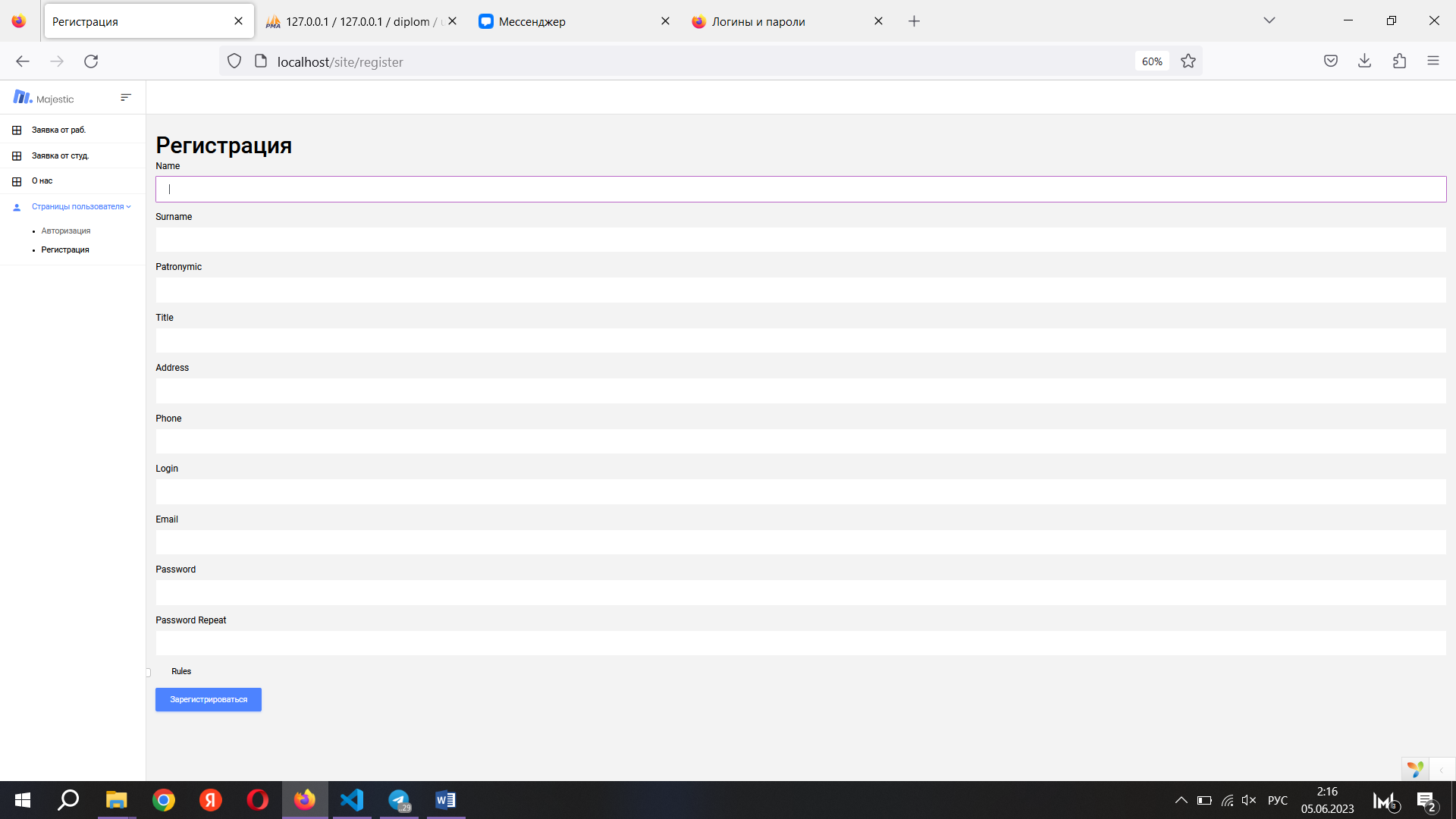


Рисунок 28 – Страница «Регистрация»

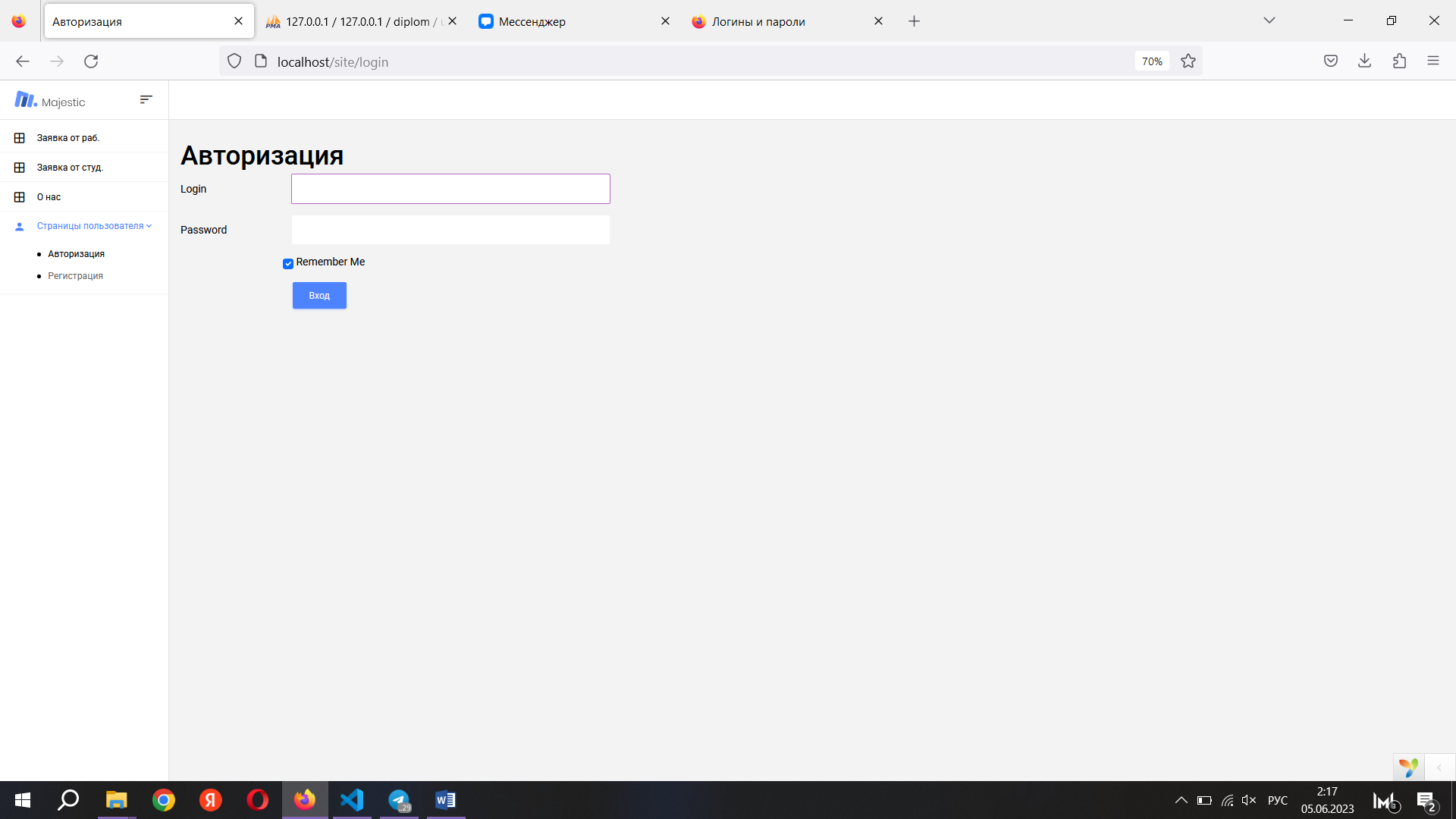


Рисунок 29 – Страница «Авторизация»

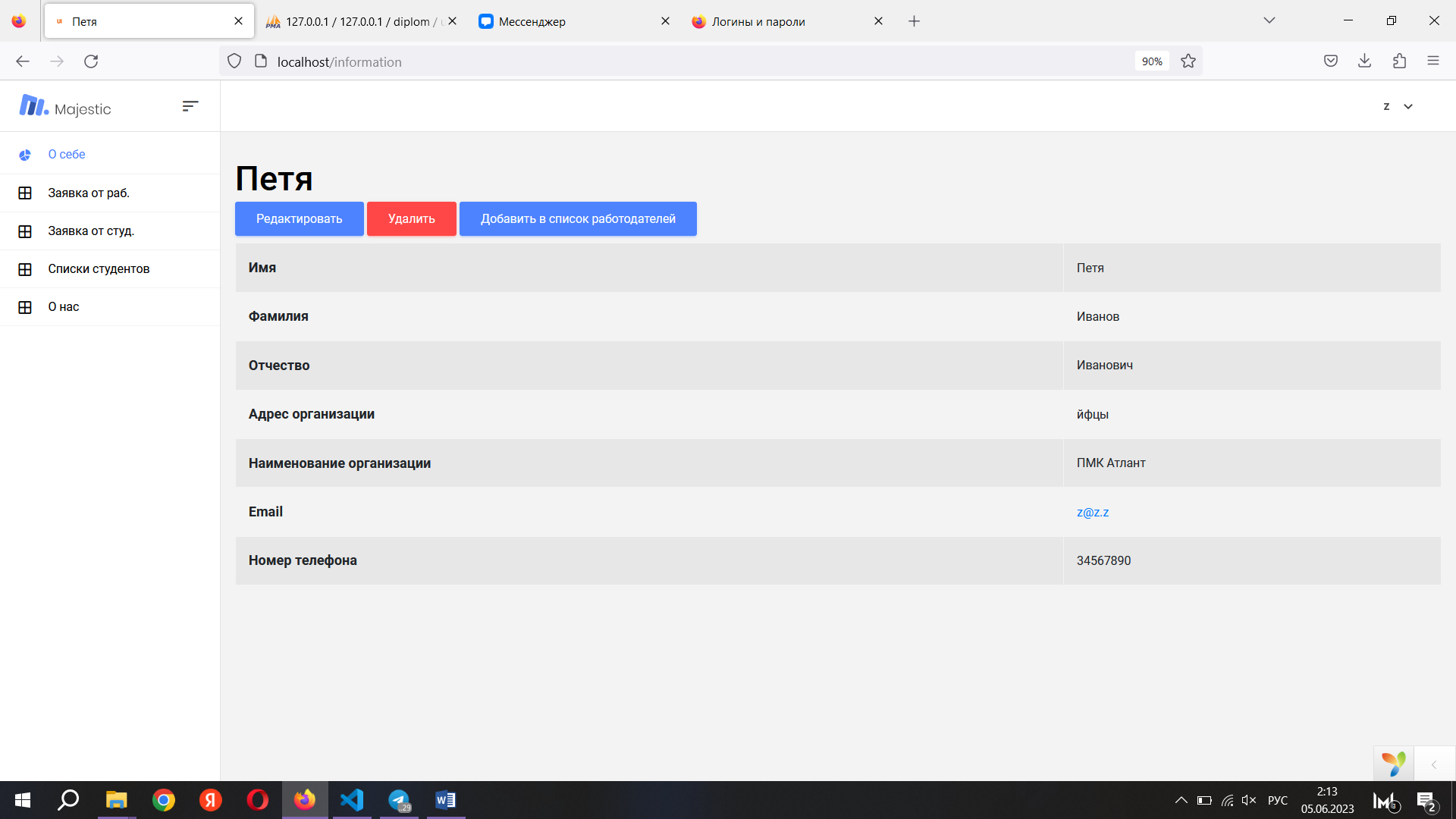


Рисунок 30 – Страница «Резюме»

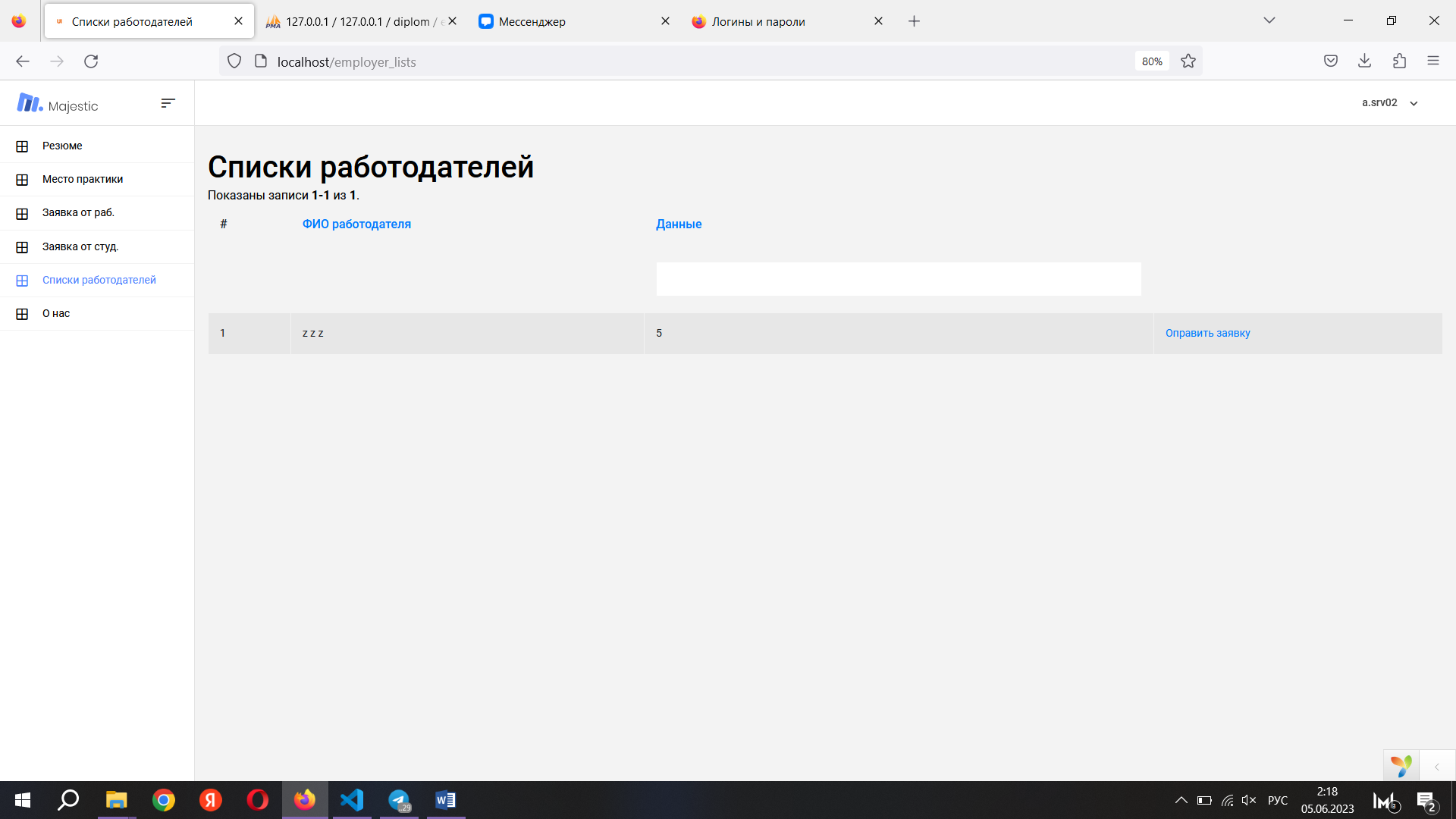


Рисунок 31 – Страница «Списки работодателей»

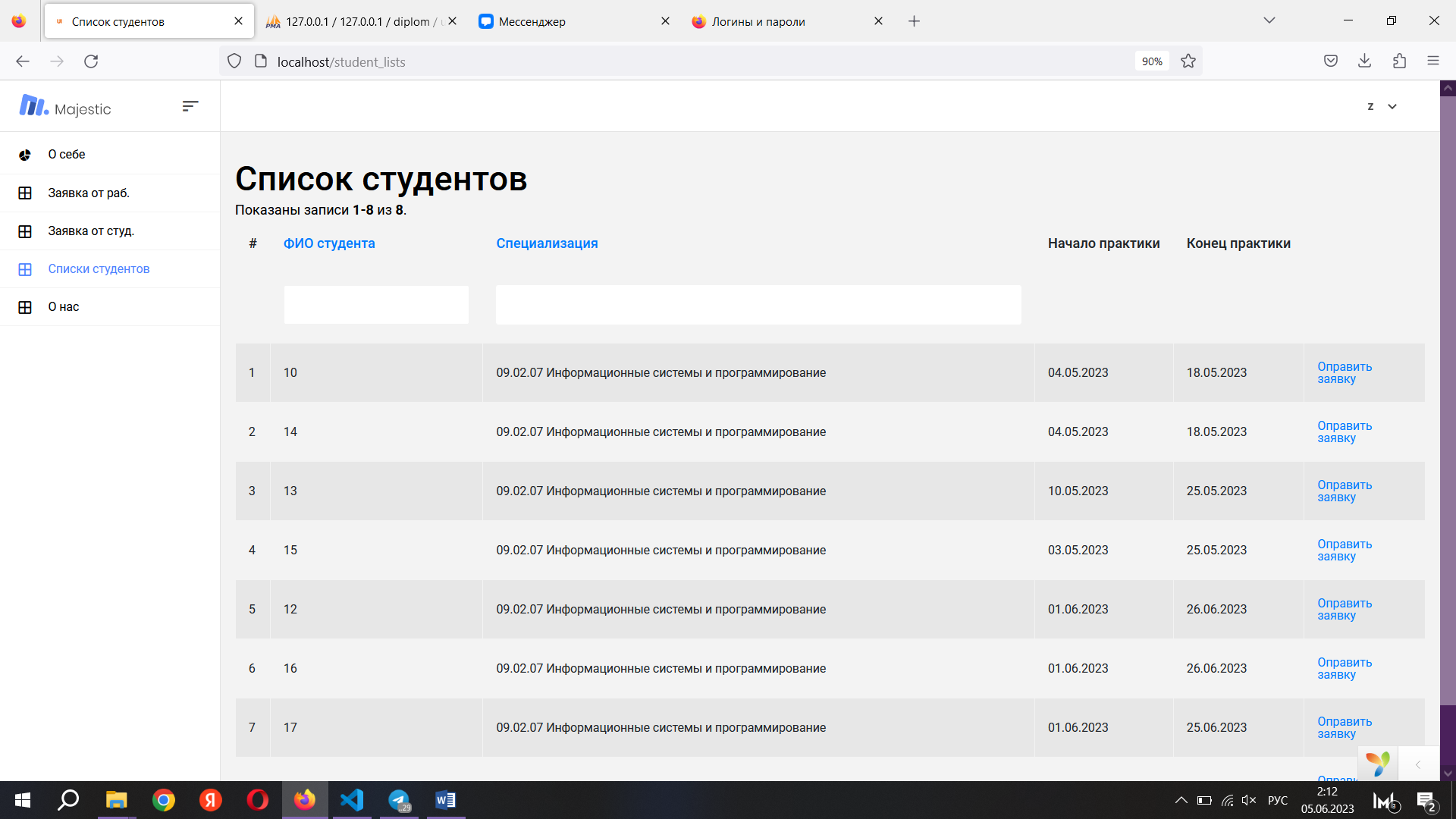


Рисунок 32 – Страница «Списки студентов»

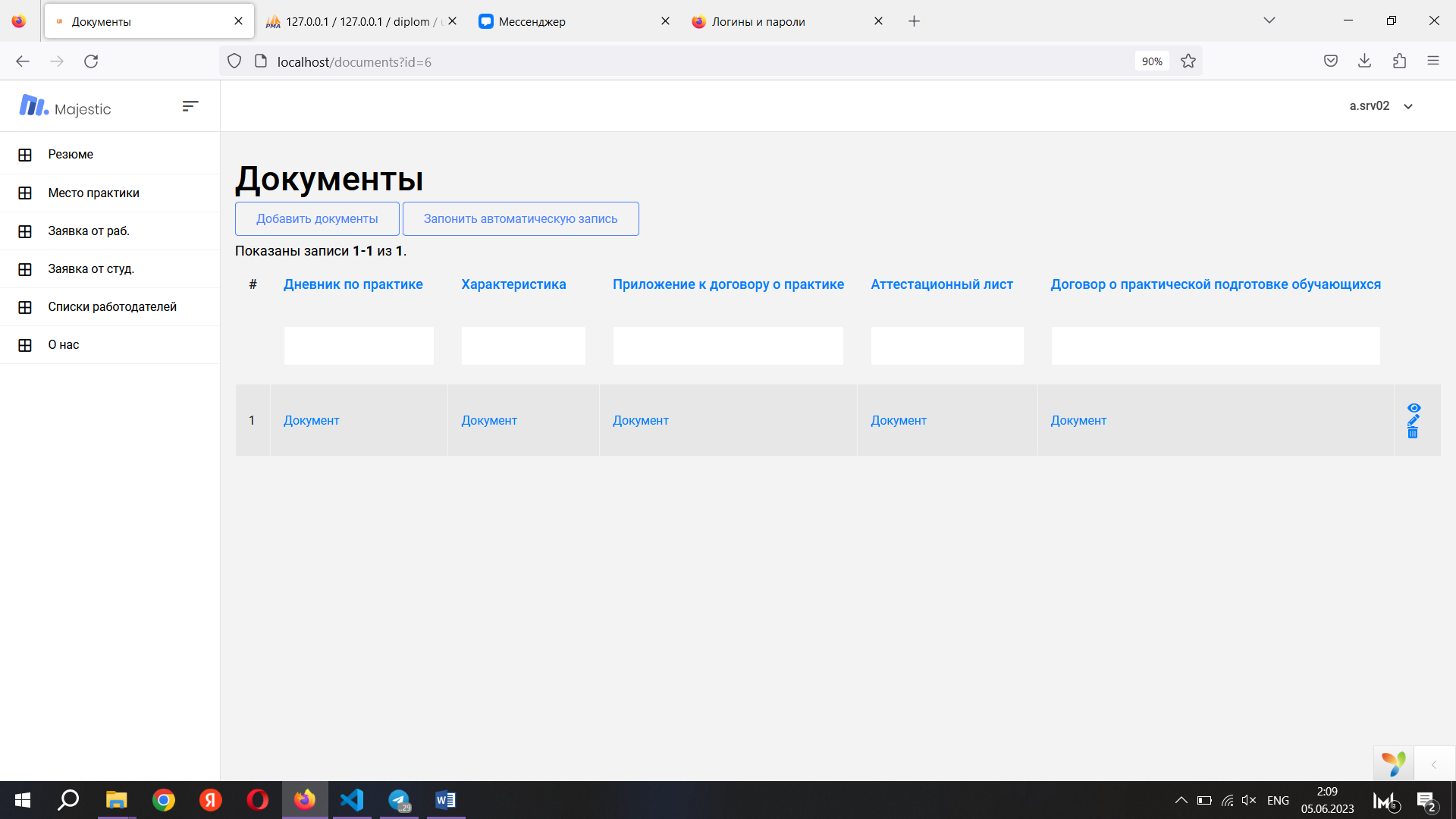


Рисунок 33 – Страница «Документы»

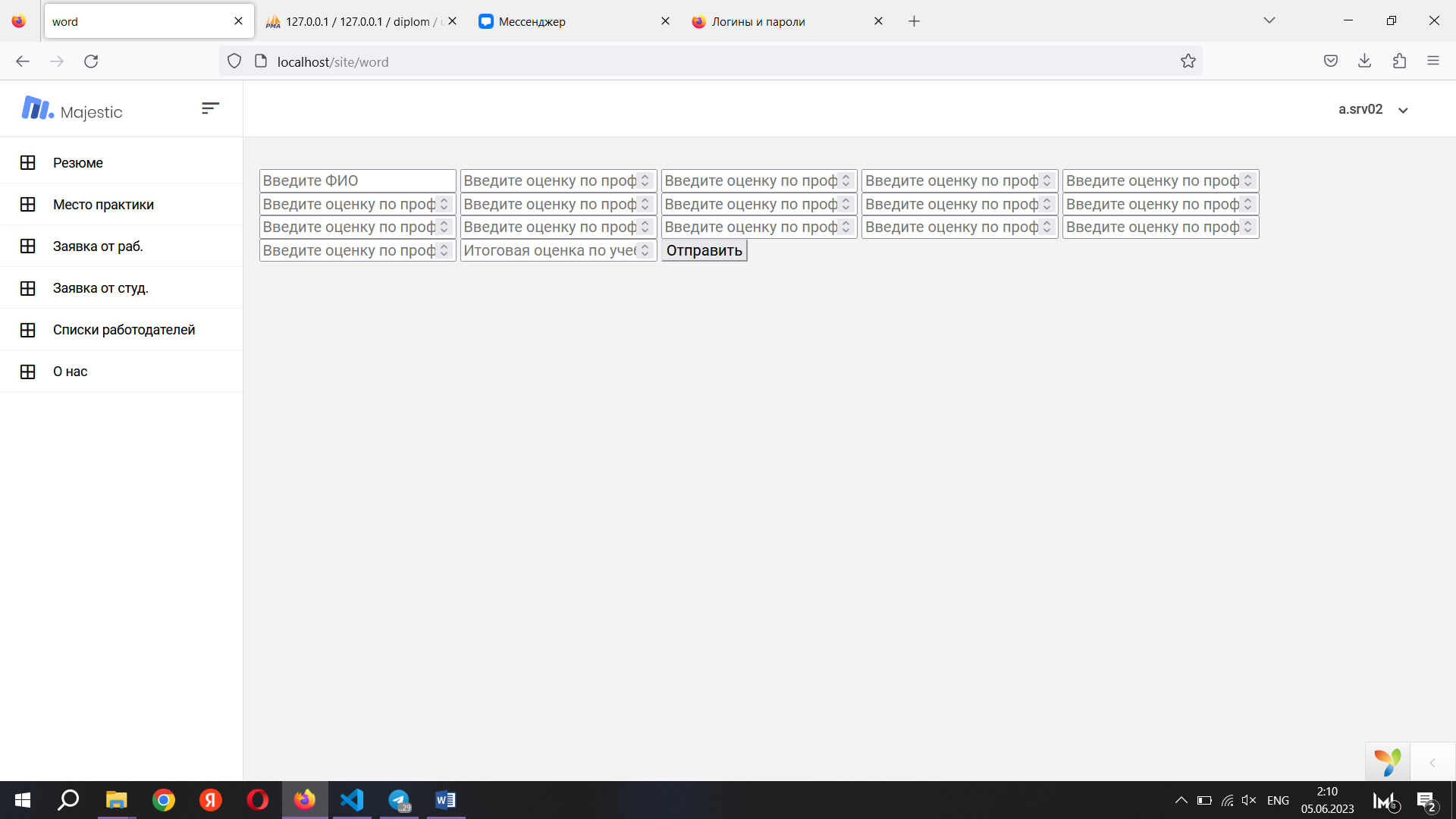


Рисунок 34 – Страница «Автоматическая запись»

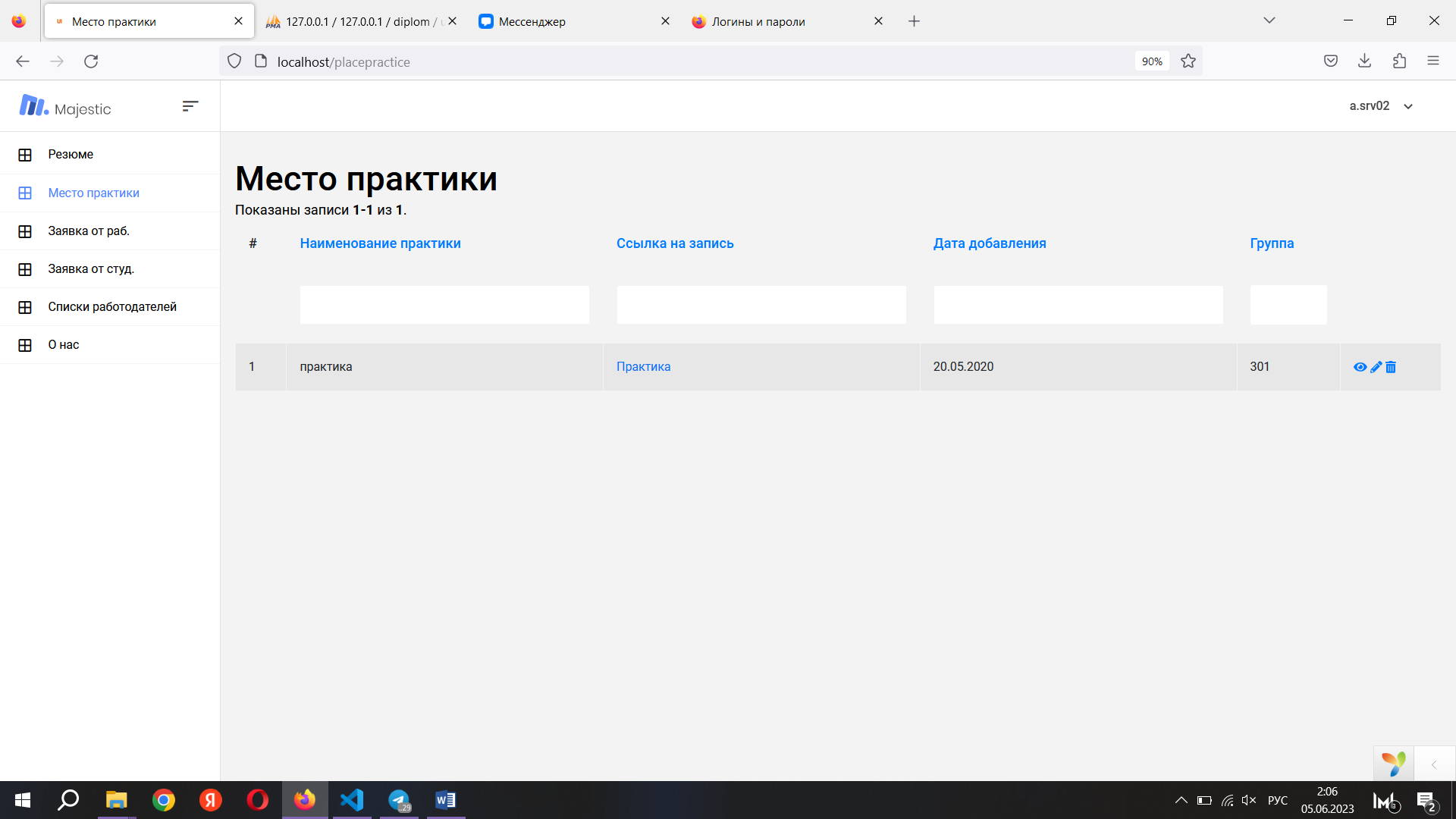


Рисунок 35 – Страница «Место практики»

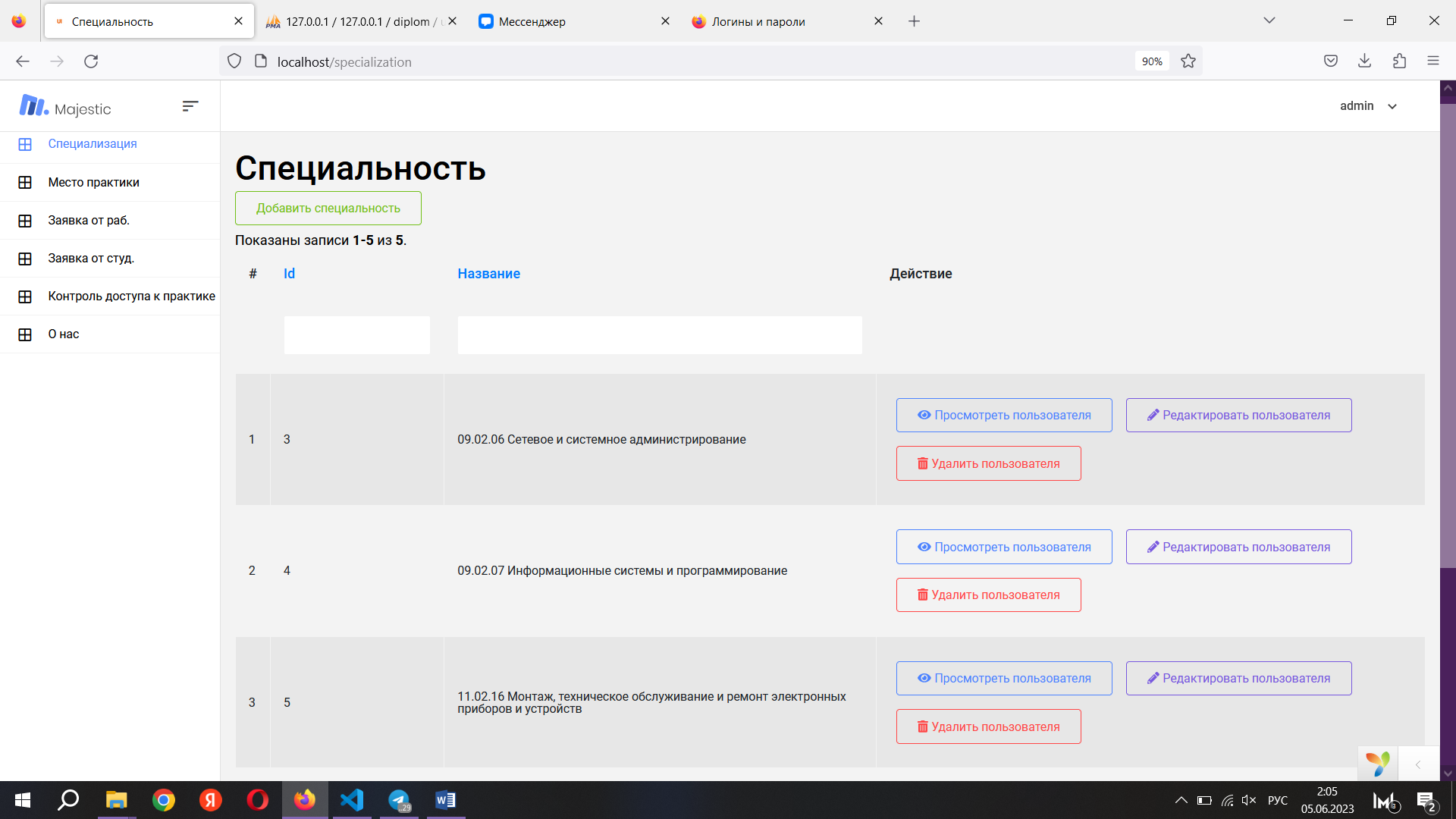


Рисунок 36 – Страница «Специальности»

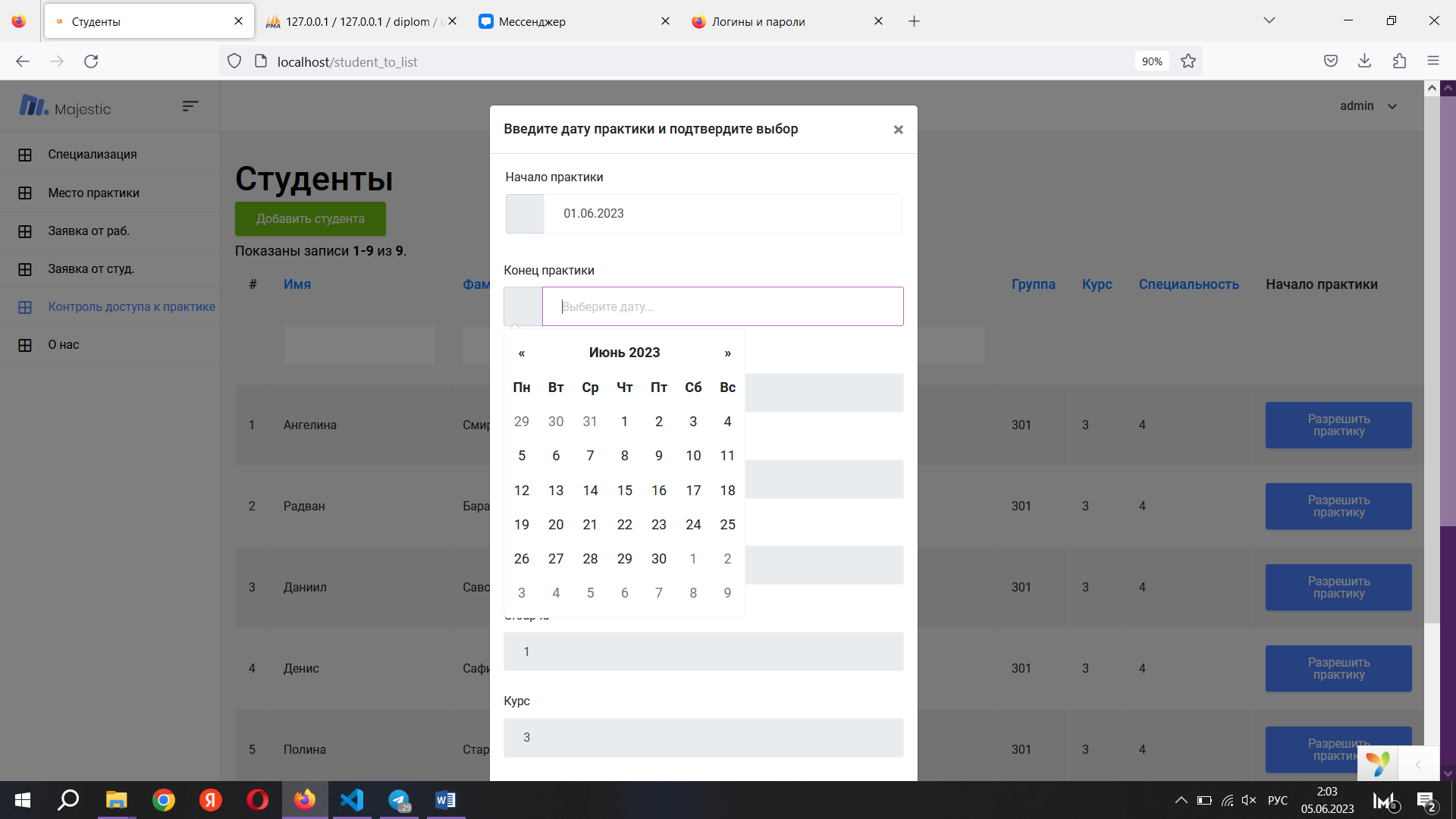


Рисунок 37 – Страница «Контроль доступа к практике»



Рисунок 38 – Страница «Добавить группу на практику»

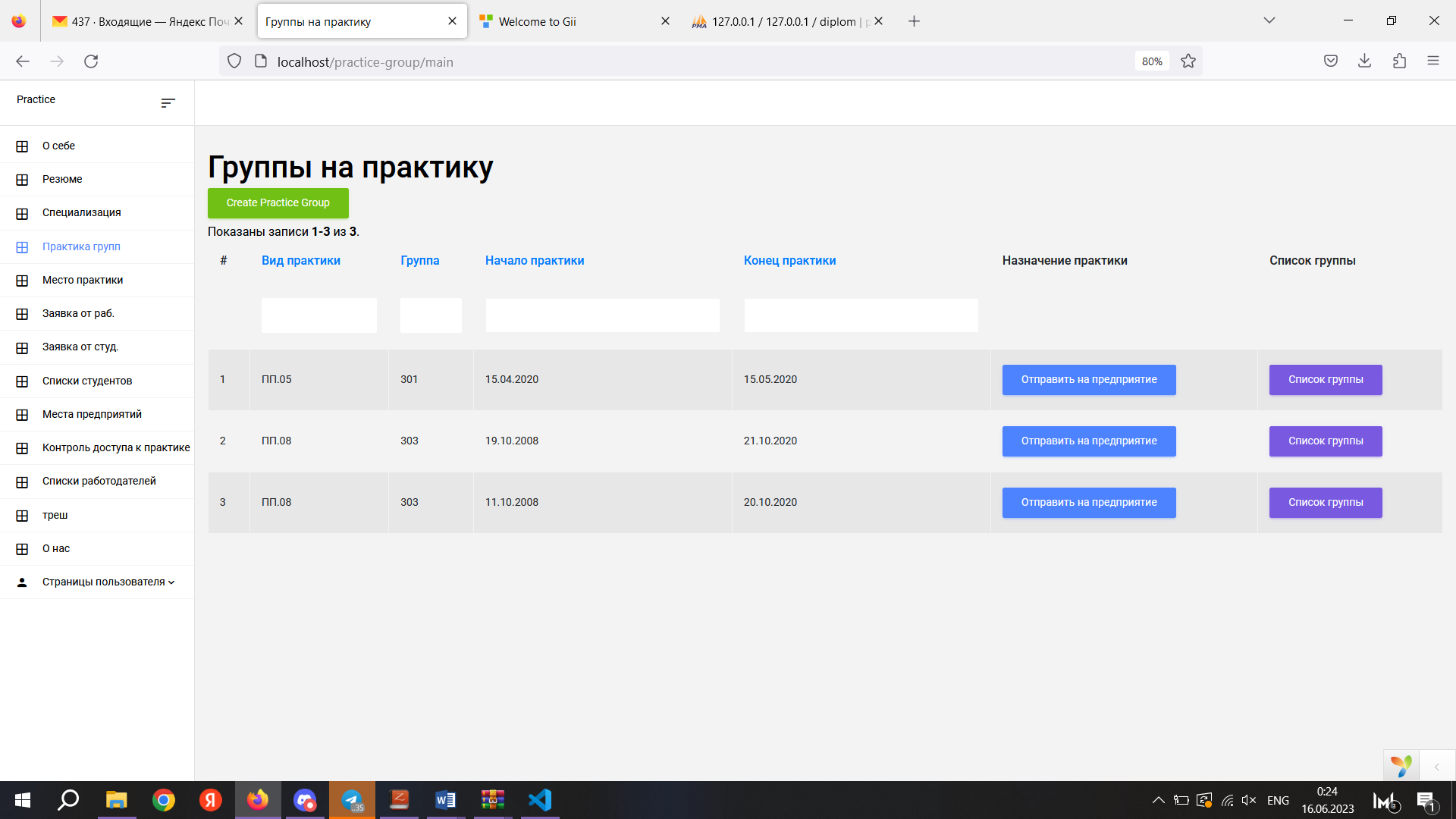


Рисунок 39 – Страница «Группы на практику»

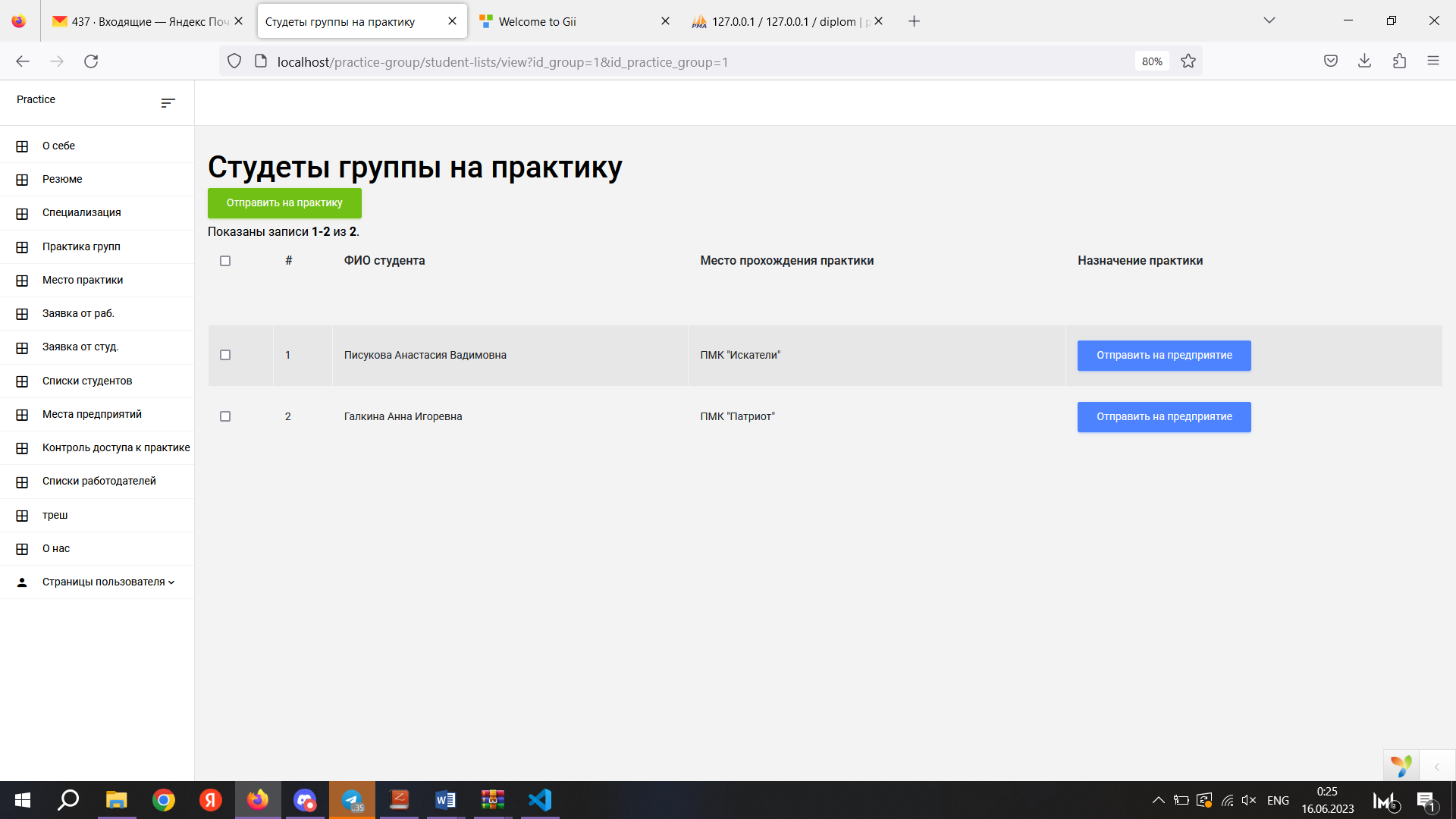


Рисунок 40 – Страница «Студенты группы на практику»

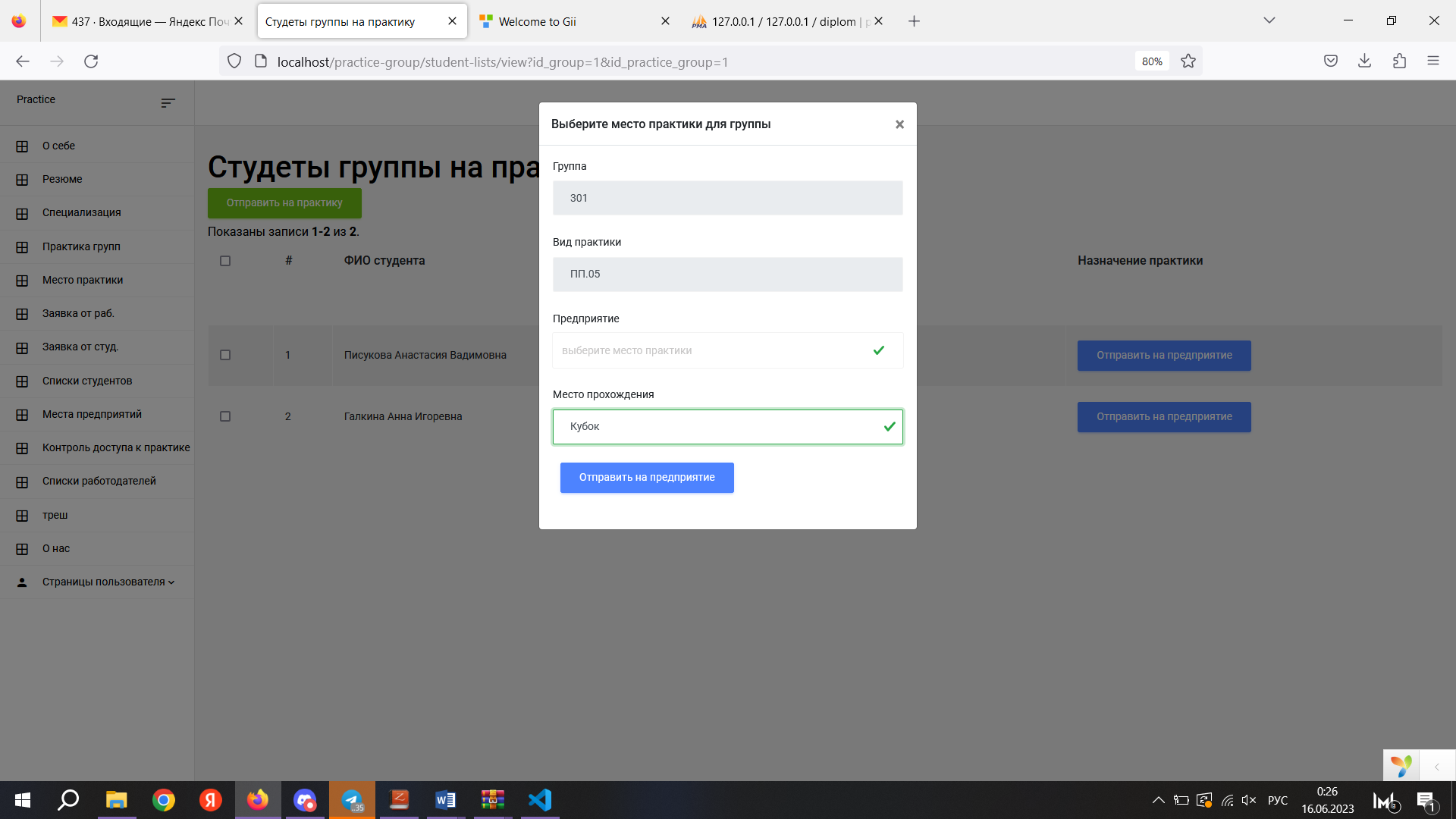


Рисунок 41 – Модальное окно

**Возможности системы.**

Всех пользователей системы можно разделить на 2 роли: пользователь и администратор. Роль пользователя делится на роли: студент и работодатель. Администратор, в свою очередь, является все тем же обычным пользователем, но выполняет иные функции по административным моментам, что расширяет его возможности.

**Руководство пользователя: «Работодатель».**

1. Регистрация и авторизация в системе.

Для того, чтобы авторизоваться или зарегистрироваться, пользователю необходимо нажать на соответствующую кнопку в навигационной панели сайта, которая изображена на рисунке 42.

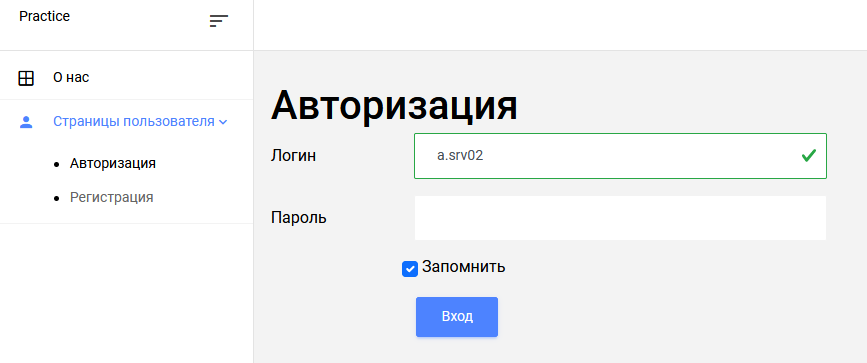


Рисунок 42 – Кнопки входа и регистрации.

После нажатия на одну из кнопок, откроется соответствующая страница, предлагающая ввести данные, необходимые для выполнения выбранного действия. Зарегистрированным пользователем будет работодатель, студентов регистрирует админ.

1. Создание резюме.

После регистрации и авторизации пользователь может заполнить своё резюме, чтобы студент мог понимать какие его ждут возможности у данного работодателя, на данном предприятии. После сохранения резюме у работодателя появляется кнопка «Добавить в список работодателей». При нажатие он автоматически отображается в списках, которые видят студенты.

1. Заявка на производственную практику.

У пользователя в навигационной панели имеет страница «Списки студентов», на данной странице работодатель может посмотреть резюме студентов, их вид практики, специальность на которую они учатся и дату практики, то есть период на который работодатель может пригласить студента.

1. Заполнение документов.

Как только студент попадает на практику к работодателю, то в конце работы он должен предоставить документы об окончании прохождения практики. В данном проект есть функция при которой работодатель может прям на сайте проставить оценки за выполненную работу и сохранить документ. При сохранении документ автоматически скачивается и можно печатать.

**Руководство пользователя: «Студент».**

1. Создание резюме.

Студентов регистрирует админ, поэтому после авторизации, необходимо заполнить резюме студента, чтобы работодатель мог видеть и выбирать подходит ли ему данный студент.

1. Выбор места практики.

В навигационной панели есть ссылка на страницу «Место практики», на данной странице студент может самостоятельно вписать своё место практики, или перейти на страницу «Список работодателей» и выбрать работодателя на понравившемся предприятии. После студент отправляет заявку работодателю и ждёт ответа. Если же у студента нет места практики и список работодателей ему не подходит, то администратор самостоятельно отправляет ученика на практику.

**Руководство пользователя: «Администратора».**

1. Добавление необходимой информации.

Администратор прописывает и создаёт важные моменты в данном проекте, такие как создание и добавление специальностей, регистрирование студентов и осуществляет доступ к практике, как по группам, так и индивидуально.

1. Отправка студентов на место практики.

Есть специальности, которые необходимо отправить на производство всей группой. Для этого админ на странице «Практика групп» может отправить всю группу сразу на практику, но есть у одного из этой группы есть уже своё место практики, то админ может индивидуально указать место практики, выбрав нужного студента и изменив данные в модельном окне, которое изображено на рисунке 43.

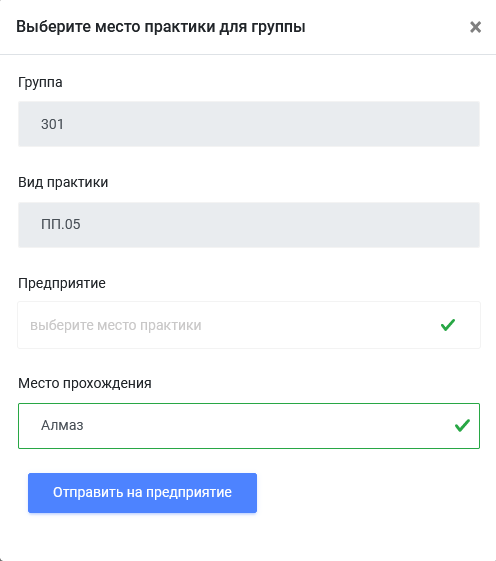


Рисунок 43 – Изменение файла .htaccess.

3.7. Руководство программиста

Для корректной работы информационной системы, ее необходимо загрузить на хостинг.

После загрузки файлов на хостинг, нужно настроить базу данных, для этого необходимо создать новую базу данных MySQL на хостинге и импортировать имеющуюся базу, если база отсутствует – создать базу, соответствующую структуре ER-диаграммы.

Для подключения базы данных необходимо редактировать файл db.php, находящийся в папке проекта config. Необходимо изменить «username», «password» и адрес хоста, эти данные получаются при создании базы данных на хостинге.

Также, для корректной работы сервиса необходимо настроить .htaccess файл в корневой папке проекта. Для этого нужно ввести код, показанный на рисунке 44.



Рисунок 44 – Изменение файла .htaccess.

## 4. ЭКОНОМИЧЕСКАЯ ЧАСТЬ

4.1. Определение затрат:

Определяют статьи затрат и нормы расхода по ним для последующего включения в расчет себестоимости программного продукта.

* *на вспомогательные материалы*, включающие: тонер для картриджа, бумага для принтера, флешка Kingston DataTraveler 100 G3 16 ГБ и т.п. (таблица 1);
* *на электроэнергию*, учитывая мощность потребляемой энергии компьютера, принтера и т.п. устройств;
* *на основную и дополнительную заработную плату* разработчика программного продукта;
* отчисления на социальные нужды;
* накладные расходы (укрупнено);

4.2. Расчет себестоимости программного продукта

Себестоимость программного продукта рассчитывается по формуле:

*С = Мвс + Э + Зозп + Здзп + Зсн+Н* ,

где - затраты на вспомогательные материалы, р.;

- затраты на электроэнергию на технологические цели, р.;

- основная зарплата разработчика, р.;

 *-* дополнительная зарплата разработчика, р.;

- взносы на социальное страхование и обеспечение, р.;

*Н -* накладные расходы, р.

Затраты на вспомогательные материалы приведены в таблице 18.

Таблица 18 - Затраты на вспомогательные материалы

|  |  |  |
| --- | --- | --- |
| Наименование затрат | Количество, шт. | Сумма, р. |
| 1. Тонер для заправки картриджей HP CF218A (№18A), HP CF230A (№18A) Тип 6.2, Bk, 50 г, банка, с воронкой-дозатором | 1 | 413 |
| 2. Бумага для офисной техники Снегурочка, А4, 80 г/кв. м, белизна 146% CIE, 500 листов | 1 | 379 |
| 3. Флешка Kingston DataTraveler 100 G316 ГБ | 1 | 469 |
| Итого: | | 1261 |

Затраты на электроэнергию рассчитываются по формуле:

,

Где

- мощность потребляемой электроэнергии, кВт;

** - стоимость одного киловатт-часа электроэнергии, р./кВт-ч;

* -* общие затраты труда на разработку программного продукта, час;

* -* коэффициент загрузки компьютера.

Подставив в формулу числовые значения вычислим затраты на электроэнергию:

*Э*=0,73·4,28·150·0,5=234, 33 *р.*

Основная заработная плата разработчика рассчитывается по формуле:

*Зозп=Счтс·Т´ОБЩ,*

где

 *-* часовая тарифная ставка разработчика, р./час;

- общие затраты труда на разработку программного продукта, час.

Подставив в формулу числовые значения вычислим основную заработную плату разработчика:

*Зозп=500\*170=85000* *р.*

Накладные расходы рассчитываются по формуле:

*Н=0,1·( Зозп+ Здзп )*,

Подставив числовые значения в формулу, получим:

*Н=* 0,1·(85000+0)=8500 *р.*

Полная себестоимость разработанного программного продукта:

*С*=1261+234, 33+85000+0+0+8500= 94995,33 *р.*

В таблице 21 сведены результаты расчетов себестоимости программного продукта.

Таблица 19 - Калькуляция себестоимости программного продукта

|  |  |
| --- | --- |
| Наименование статей расходов | Затраты, р. |
| 1 | 2 |
| 1. Основная заработная плата разработчика | 85000 |
| 2. Затраты на электроэнергию | 234, 33 |
| 3 Накладные расходы | 8500 |
| 4. Вспомогательные материалы | 1261 |
| 5. Полная себестоимость программного  продукта | 94995,33 |

Вывод: Себестоимость программного продукта составила 94995 рублей 33 копейки.

## ЗАКЛЮЧЕНИЕ

В процессе написания дипломной работы была реализована инфор-мационная система, позволяющая записывать, хранить а также искать ме-ста производственной практики. Тема была доработана и в окончательной версии у пользователей появилась возможность делать отклики на заявки. Это сделало систему еще более удобной.

Также была успешно исследована предметная область, а также изу-чены различные аналоги, на основе которых была разработана информаци-онная-система .

Были созданы компоненты, прототипы страниц и макеты страниц сайта.

Все поставленные задачи, а также цель работы были успешно выпол-нены в рамках выпускной квалификационной работы.

Разработанная информационная система имеет следующие преимущества:

* учитывает все нюансы и полностью обеспечивает лёгкое заполнение данных на сайте;
* не требует инсталляции как самой информационной системы, так и дополнительных программных продуктов, библиотек и драйверов;
* не зависит от программного обеспечения рабочих мест сотрудников и функционирует в полном объеме на любом компьютере с любой версией ОС Windows;
* не требует никаких изменений, корректировок и доработок при смене местоположения информационной системы на базовом компьютере;
* обеспечивает многопользовательский режим работы аналогично архитектуре «клиент-сервер» без использования отдельных клиентских приложений;
* обеспечивает защиту персональных данных всех пользователей системы;
* имеет низкую себестоимость, благодаря удачно найденным решениям, значительно сократившим объем программного кода и время разработки.

Разработанная информационная система может служить основой для дальнейшей автоматизации информационных процессов при распределении студентов на практику.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

**Электронные ресурсы:**

1. Официальный сайт справочной документации по языку Php. - URL: <https://www.php.net/> (дата обращения: 21.04.2023).
2. Официальный сайт документации фреймворка Yii. - URL: <https://www.yiiframework.com/> (дата обращения: 21.04.2023).
3. Официальный сайт документации MySQL базы данных. - URL: <https://dev.mysql.com/> (дата обращения: 13.05.2023).
4. Официальный сайт документации Bootstrap. - URL: <https://bootstrap-4.ru/> (дата обращения: 13.05.2023).
5. Официальный сайт документации Visual Studio Code. - URL: <https://code.visualstudio.com/> (дата обращения: 20.04.2023).
6. Официальный сайт документации Open Server Panel. - URL: <https://ospanel.io/> (дата обращения: 20.04.2023).
7. Методы авторизации Yii. Настройка RBAC Manager. - URL: <https://www.yiiframework.com/doc/guide/2.0/ru/security-authorization> (дата обращения: 20.04.2023).
8. Расширения для Yii Framework 2.0. Виджет highcharts. - URL: <https://www.yiiframework.com/extension/yii2-highcharts-widget> (дата обращения: 19.05.2023).
9. Официальный сайт документации Krita 5.0. - URL: <https://docs.krita.org/en/index.html> (дата обращения: 17.05.2023).
10. James R. Bailey and Scheherazade Rehman Don’t Underestimate the Power of Self-Reflection: статья научно-популярного журнала Harvard Business Review. - URL: <https://hbr.org/2022/03/dont-underestimate-the-power-of-self-reflection?autocomplete=true> (дата обращения: 24.05.2023).
11. Бесплатный шаблон Bootstrap «Majestic Admin». - URL: https://bootstraptema.ru/stuff/templates\_bootstrap/admin/majestic\_admin/4-1-0-5501 (дата обращения: 09.04.2023).
12. Composer. A Dependency Manager for PHP. — URL: https://getcomposer.org. (дата обращения: 08.05.2023).
13. Расширения для Yii Framework 2.0. - URL: https://demos.krajee.com/ (дата обращения: 17.05.2023).

## ПРИЛОЖЕНИЕ А (обязательное) Программные коды

Метод в StudentListsController, реализует вывод групп и вид практик.

public function actionView($id\_group,$id\_practice\_group)

{

// var\_dump($id\_group,$id\_practic); die;

if( !StudentPractice::getStudentPracticCount($id\_group,$id\_practice\_group) ) {

StudentPractice::addStudentPracticGroup($id\_group,$id\_practice\_group);

}

$searchModel = new StudentPracticeSearch();

$dataProvider = $searchModel->search($this->request->queryParams);

$viewPractice = ViewPractice::getViewPracticeList();

$organization = Organization::getOrganization();

return $this->render('view', [

'searchModel' => $searchModel,

'dataProvider' => $dataProvider,

'viewPractice' => $viewPractice,

'organization' => $organization,

]);

}

Метод в StudentListsController, реализует сохранение, перенос и автоматическое заполнение данных на другой странице.

public function actionSet($id)

{

if( Yii::$app->request->isPost) {

if( ($model = StudentPractice::findOne($id)) && $model->load(Yii::$app->request->post()) ) {

// VarDumper::dump($model->attributes, 10, true); die;

if( !empty($model->organization\_id) || !empty($model->place\_title) ) {

$query = 'update student\_practice '

. "set "

. (empty($model->organization\_id) ? 'place\_title' : 'organization\_id') . " = "

. (empty($model->organization\_id) ? "'{$model->place\_title}'" : $model->organization\_id)

. (empty($model->place\_title) ? ', place\_title = NULL' : '')

. ' where '

. "id = {$model->id} "

;

$num = Yii::$app->db->createCommand($query)->execute();

}

return $this->redirect(['view', 'id\_group' => $model->practiceGroup->group->id, 'id\_practice\_group' => $model->practice\_group\_id]);

}

} else {

ECHO '2'; DIE;

}

}

Метод в StudentListsController, реализует вывод организации, групп и вид практик.

public function actionSetStudentsSelect()

{

$model = new StudentPractice();

if( Yii::$app->request->isPost && Yii::$app->request->isAjax ) {

if( $model->load(Yii::$app->request->post())) {

$model->save();

return $this->redirect(['view', 'id\_group' => $model->practiceGroup->group->id, 'id\_practice\_group' => $model->practice\_group\_id]);

}

} else {

$organization = Organization::getOrganization();

$viewPractice = ViewPractice::getViewPracticeList();

$group = PracticeGroup::getGroupList();

return $this->renderAjax('modal-group', compact('model', 'group', 'organization', 'viewPractice'));

}

}

public function actionGetStudentList($group\_id)

{

$items = User::getStudentList($group\_id);

Yii::$app->response->format = 'json';

return $items;

}

PHPW подключен через composer имеет данный вид html-кода.

<?php

/\*\* @var yii\web\View $this \*/

use yii\bootstrap5\Html;

$this->title = 'word';

$this->params['breadcrumbs'][] = $this->title;

?>

<div class="word">

<form action="/site/word-save" method="POST" enctype="multipart/form-data">

<input id="form-token" type="hidden" name="<?=Yii::$app->request->csrfParam?>"

value="<?=Yii::$app->request->csrfToken?>"/>

<!-- <input type="date" name="birth"> -->

<input type="text" name="name" placeholder="Введите ФИО">

<input type="number" name="value\_1" placeholder="Введите оценку по профессиональные компетенции №1">

<input type="number" name="value\_2" placeholder="Введите оценку по профессиональные компетенции №2">

<input type="number" name="value\_3" placeholder="Введите оценку по профессиональные компетенции №3">

<input type="number" name="value\_4" placeholder="Введите оценку по профессиональные компетенции №4">

<input type="number" name="value\_5" placeholder="Введите оценку по профессиональные компетенции №5">

<input type="number" name="value\_6" placeholder="Введите оценку по профессиональные компетенции №6">

<input type="number" name="value\_7" placeholder="Введите оценку по профессиональные компетенции №7">

<input type="number" name="value\_8" placeholder="Введите оценку по профессиональные компетенции №8">

<input type="number" name="value\_9" placeholder="Введите оценку по профессиональные компетенции №9">

<input type="number" name="value\_10" placeholder="Введите оценку по профессиональные компетенции №10">

<input type="number" name="value\_11" placeholder="Введите оценку по профессиональные компетенции №11">

<input type="number" name="value\_12" placeholder="Введите оценку по профессиональные компетенции №12">

<input type="number" name="value\_13" placeholder="Введите оценку по профессиональные компетенции №13">

<input type="number" name="value\_14" placeholder="Введите оценку по профессиональные компетенции №14">

<input type="number" name="value\_15" placeholder="Введите оценку по профессиональные компетенции №15">

<input type="number" name="grading\_general" placeholder="Итоговая оценка по учебной/производственной практике">

<button type="submit">Отправить</button>

</form>

</div>