ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

**АННОТАЦИЯ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Студент | | Бученков Е.Е. | | | | | |
| (Фамилия, И.О.) | | | | | | | |
| Факультет | ИКТ | | | | | | |
| Кафедра | ПС | | | | | Группа | K3122 |
| Направление (специальность) | | | | | Инфокоммуникационные технологии и системы связи | | |
| Руководитель | | | Лямин А.В., Университет ИТМО, доцент ф-та ПИиКТ, доцент, к.т.н. | | | | |
| ( Фамилия, И.О., место работы, должность, ученое звание, степень ) | | | | | | | |
| Дисциплина | | Информатика | | | | | |
|  | | | | | | | |
| Наименование темы | | | | «Система учета учебных достижений студентов кафедры» | | | |
|  | | | |  | | | |

**ХАРАКТЕРИСТИКА КУРСОВОГО ПРОЕКТА (РАБОТЫ)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1. Цель и задачи работы** | Предложены студентом | | | Сформулированы при участии студента | |
| Определены руководителем | |
| Требуется разработать модуль системы учета учебных достижений студентов кафедры и структуру JSON- | | | | | |
| файла, содержащего данные объектов системы. Для каждой кафедры необходимо хранить ее название, дату | | | | | |
| основания, информацию об университете, к которому принадлежит кафедра, данные о местоположении, | | | | | |
| краткое описание, информацию о ее заведующем, включающую его имя, фамилию, отчество при наличии и | | | | | |
| дату рождения, список групп, содержащих информацию о названии группы и списке студентов, | | | | | |
| характеризующихся краткой информацией, включающей имя, фамилию, отчество при наличии, дату | | | | | |
| рождения, а также списком достижений, для каждого из которых указывается название, категория, дата | | | | | |
| достижения и краткое текстовое описание. Модуль должен предоставлять возможность получения данных о | | | | | |
| списке кафедр, хранящихся в базе данных, возможность перехода к интерфейсу просмотра краткой | | | | | |
| информации о списке достижений всех студентов кафедры, интерфейсу просмотра подробной информации об | | | | | |
| отдельном достижении конкретного студента. | | | | | |
| **2. Характер работы** | | | | | |
|  | Расчет | | | Конструирование | |
|  | Моделирование | | | Другое, | |
| Разработка ПО | |
| **4. Содержание работы** | |  | | | |
| Введение; | | | | | |
| 1. Анализ и уточнение задания; | | | | | |
| 2. Инфологическая модель базы данных; | | | | | |
| 3. Описание ролей и интерфейсов пользователей системы; | | | | | |
| 4. Описание архитектуры системы; | | | | | |
| 5. Описание прогрммного кода системы; | | | | | |
| Заключение; | | | | | |
| **5. Выводы** | | |  | |  |
| Создана база данных, хранящая в себе информацию о кафедрах и достижениях студентов, а также | | | | | |
| информацию о самих студентах, преподавтелях и модераторах. Реализован модуль системы учета учебных | | | | | |
| достижений студентов кафедры, хранящихся в базе данных. | | | | | |

|  |  |  |
| --- | --- | --- |
| Руководитель | |  |
| подпись, дата | | |
| Студент |  | |
| подпись, дата | | |

**Содержание**

[Введение 6](#_Toc532428958)

[1. Анализ и уточнение задания 7](#_Toc532428959)

[2. Инфологическая модель базы данных 9](#_Toc532428960)

[3. Описание ролей и интерфейсов пользователей системы 10](#_Toc532428961)

[4. Описание архитектуры системы 12](#_Toc532428962)

[5. Описание программного кода системы 15](#_Toc532428963)

[Заключение 21](#_Toc532428964)

[Список используемых источников 22](#_Toc532428965)

**Введение**

Была поставлена цель разработать программное обеспечение для учета учебных достижений студентов кафедры. Данное ПО подразумевает собой веб-сайт. На главной странице этого сайта необходимо для каждой кафедры вывести название, дату основания, университет, к которому принадлежит данная кафедра, местоположение, краткое описание и ФИО руководителя. Каждый пользователь системы характеризуется id, никнеймом, паролем и группой безопасности ("студент", "преподаватель", "модератор", "заблокирован"). Каждое достижение студента характеризуется id, наименованием, автором, описанием и датой добавления в систему.

Зарегестрированные пользователи имеют возможность авторизации в сиситеме, добавление своей фотографии в профиль, добавление нового достижения, добавление/изменение/удаление пользователей и одобрение/отклонение достижений других пользователей.

**1. Анализ и уточнение задания**

На данном этапе курсовой работы необходимо было уточнить техническое задание проекта, а также определить аналоги разрабатываемой системы. Уточнение технического задание подразумевало собой уточнение технологий, которые будут использоваться в разработке самой системы, а также ряд вопросов относительного того, что будет представлять из себя данная система (например: Как должны отображаться достижения студентов? Какие группы безопасности должны присутствовать в проекте? и тд.).

Что касается аналогов, то были рассмотрены 3 варианта, их плюсы и минусы, использованные для разработки технологии, а также краткое описание функционала:

1. **ИНФОРМАЦИОННАЯ СИСТЕМА УЧЕТА УСПЕВАЕМОСТИ СТУДЕНТОВ ЗАОЧНОЙ ФОРМЫ ОБУЧЕНИЯ**

Использованные технологии: Delphi.

[Документация: <http://technology.snauka.ru/2014/12/5239>]

Функционал:

* формирование сведений о студенте;
* формирование групп студентов с учетом рабочих учебных планов;
* гибкий учет изменений в учебных планах направлений или специальностей;
* учет успеваемости студента в течение сессии;
* контроль студентов, имеющих задолженности по сессиям;
* регулирование успеваемости студента в течение срока обучения;
* учет успеваемости студента по аттестации за сессию;
* поиск студентов по заданным параметрам;
* формирование списка студентов, имеющих задолженности (с указанием дисциплины и преподавателя);
* формирование списка студентов, претендующих на красный диплом.

Плюсы: Поддерживается добавление функционала. Поддерживается поиск по фильтрам (ФИО, место жительства, результаты сессии и т.д.).

Минусы: Является десктопным приложением. Не совсем удобный и понятный интерфейс из-за большого количества кнопок.

1. **ЦЕНТР ДИСТАНЦИОННОГО ОБУЧЕНИЯ УНИВЕРСИТЕТА «ИТМО»**

Использованные технологии: XHTML, Java, CSS.

[Документация: <https://de.ifmo.ru/--doc/tz_63.pdf>]

Функционал:

* Отображение баллов студента по учебным дисциплинам;
* Отображение достижений студента;
* Формирование рейтинга студентов;
* Личный кабинет (далее «ЛК») у каждого студента;
* Возможность загружать лекции на портал;
* Присутствует чат/форум.

Плюсы: Привязка к AcademicNT. Удобный интерфейс. Студентам доступен электронный архив конспектов по учебным дисциплинам.

Минусы: Задержки в подключении учебных дисциплин к ЦДО. Отсутствует оформление.

1. **ЛИЧНЫЙ КАБИНЕТ СТУДЕНТА МГТУ ИМ. БАУМАНА**

Среда программирования: Bootstrap, JavaScript(Библиотека: JQuerry), CSS, HTML.

[Документация: <https://www.kabinetpro.ru/obrazovanie/bmstu.html>]

Функционал:

* Предоставление данных об успеваемости, отслеживании сдачи экзаменов и курсовых, студент может узнать перевели его на следующий курс;
* Предоставление учебной литературы и необходимых методических материалов;
* Отображение информации о достижениях студентов в их ЛК;
* Отображение новостей об аттестации, соревнованиях, турнирах и мероприятиях на главной странице.

Плюсы: Приятный и удобный интерфейс. Присутствует оформление. ЛК у каждого студента.

Минусы: Обилие вложенных меню и кнопок, из-за чего пользователю трудно ориентироваться на сайте.

По итогу выше приведённых аналогов разрабатываемой системы было решено, что разрабатываемая система получит следующий функционал:

* Формирование сведений о студенте (фото студента, никнейм, группа безопасности);
* Отображение достижений студента (с полным описанием);
* Отображение информации об учебных кафедрах;
* Отображение полного листа всех студентов кафедры (только для пользователей с группой безопасности "Модератор").

После данного этапа было определено, что система будет разрабатываться c использованием следующих технологий: HTML, CSS, Bootstrap, Python (Фреймворк: Django).

**2. Инфологическая модель базы данных**

На втором этапе была разработана инфологическая модель базы данных проекта. Были заданы необходимые сущности, их атрибуты, а также их структура и описание. Данные сущности представлены в *таблице 1.*

*Таблица 1. Объекты базы данных*

|  |  |
| --- | --- |
| Сущность | Описание сущности |
| Cafedras | Содержит информацию о кафедрах (наименование, дата основания, университет, местоположение, краткое описание, информация о руководителе (ФИО, дата рождения). |
| Achieves | Содержит информацию о достижениях студентов (ID, наименование, дата публикации на сайте, автор (может быть только студент), проверяющий (может быть только преподаватель), описание) |
| Users | Содержит информацию о пользователях (ID, никнейм, пароль, группа безопасности, фотография). |

Для визуализации данной таблицы приведен следующий *рисунок 1.* С JSON-файлами базы данных ("users.json", "achieves.json", "cafedras.json") можно ознакомиться в папке "data".

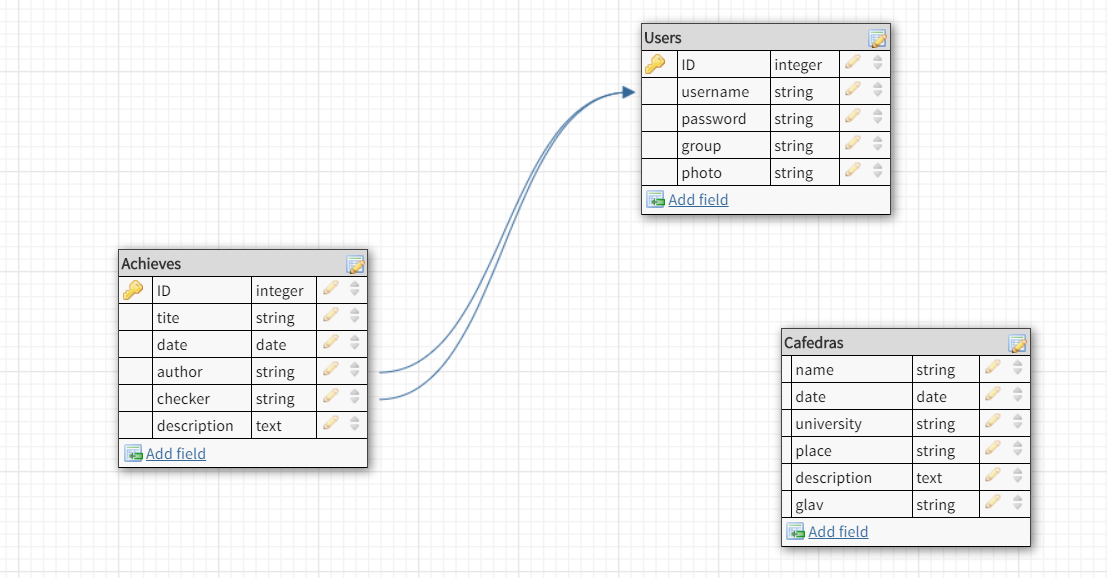


Рисунок 1 - Структура базы данных

**3. Описание ролей и интерфейсов пользователей системы**

Было решено создать следующие роли: "Модератор", "Преподаватель", "Студент", "Заблокирован".

1. Студент:

a. Имеет возможность авторизоваться на сайте;

b. Может менять свою фотографию;

c. Может добавлять новые достижения в систему;

d. Может просматривать страницу с кафедрами и достижениями всех студентов.

2) Преподаватель:

a. Имеет возможность авторизоваться на сайте;

b. Может менять свою фотографию;

c. Может одобрять/отклонять достижения студентов;

d. Может просматривать страницу с кафедрами и достижениями всех студентов.

3) Модератор:

a. Имеет возможность авторизоваться на сайте;

b. Может менять свою фотографию;

c. Может удалять отклонённые преподавателями достижения студентов;

d. Может добавлять новых пользователей, а также изменять группы/блокировать/ удалять существующих пользователей.

4) Заблокированный:

a. Не может авторизоваться на сайте, вследствие чего не имеет доступа к системе.

Некоторые интерфейсы:

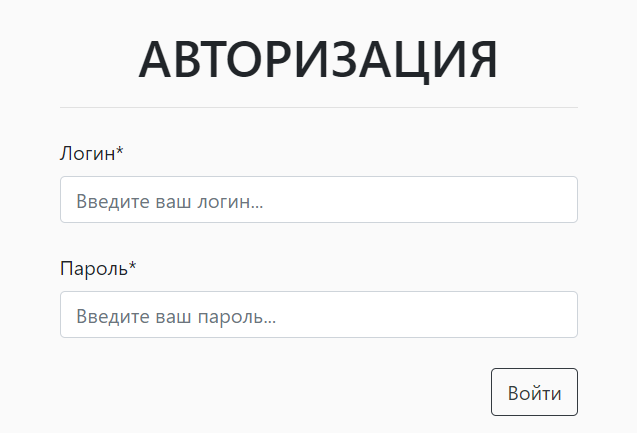


Рисунок 2 - Интерфейс авторизации

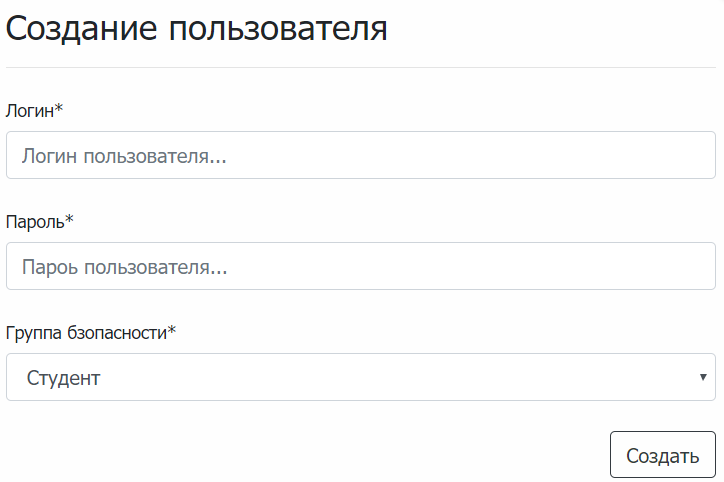


Рисунок 3 - Интерфейс добавления пользователя (только для Модератора)

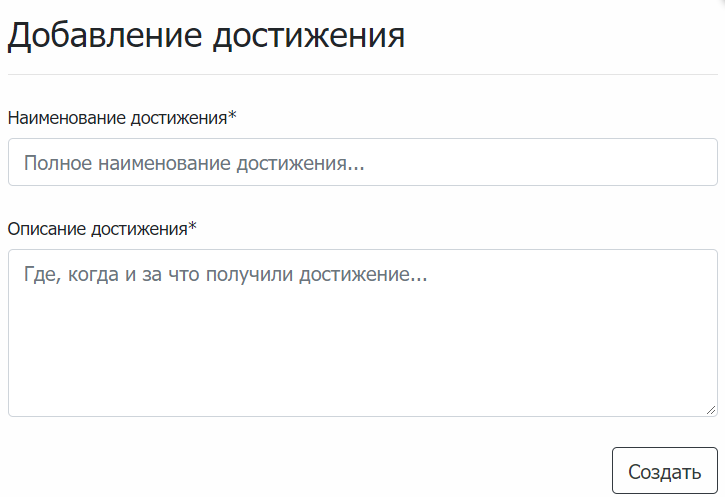


Рисунок 4 - Интерфейс добавления достижения (только для Студентов)

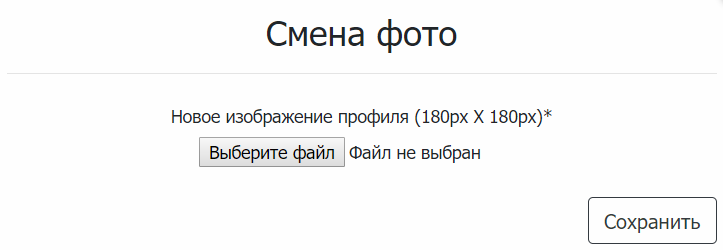


Рисунок 5 - Интерфейс смены фотографии

**4. Описание архитектуры системы**

Структура архитектуры системы представлена, а подробное описание файлов представлено в *таблице 2* (все пути считаются от корневой папки проекта)*.*

*Таблица 2. Архитектура системы*

|  |  |  |
| --- | --- | --- |
| Каталог | Имя файла | Описание |
| \main\templates\main\ | index.html | Основной шаблон, от которого наследуются все остальные, тут описан хедер, профиль пользователя и положение контента. |
|  | auth.html | Шаблон авторизации, здесь отключены все ссылки хедера. Выводит форму входа. |
|  | achieve.html | Шаблон отображения достижений всех студентов. |
|  | home.html | Шаблон стартовой страницы для авторизованных пользователей. Выводит информацию о кафедрах. |
|  | manage.html | Шаблон выводящий панель управления пользователями, к которому имеют доступ только модераторы. Выводит список всех пользователей и кнопки управления. |
| \www\ | settings.py | Файл для настройки проекта (объявление приложений, сторонних модулей и тд.). |
|  | urls.py | Файл для подключения модулей. |
|  | wsgi.py | Файл стандарта развертывания веб-приложений на Django. |
| \main\ | forms.py | Файл создания форм (поля и прочее, через что пользователь может вводить данные). |
|  | models.py | Файл моделей, в котором происходит работа с базой данных (JSON файлами). |
|  | urls.py | Файл, отвечающий за отображение определенных страниц по определённому адресу. |
|  | views.py | Файл, в котором содержится логика проекта, а также подключение шаблонов. |
| \main\data\ | achieves.json | JSON файл, в котором содержатся сущности достижений студентов. |
|  | cafedras.json | JSON файл, в котором содержатся сущности кафедр. |
|  | users.json | JSON файл, в котором содержатся сущности пользователей. |
| \main\static\main\ | styles.css | Файл стилей (большинство стилей подключены в шаблоне index.html). |

*Таблица 2*

|  |  |  |
| --- | --- | --- |
| \media\ |  | Каталог, в котором хранятся фотографии пользователей. |

Рассмотрим более детально структуру каждого json файла:

1) Структура JSON файла с достижениями студентов - achieves.json:



Рисунок 6 - achieves.json

2) Структура JSON файла со всеми пользователями - users.json:

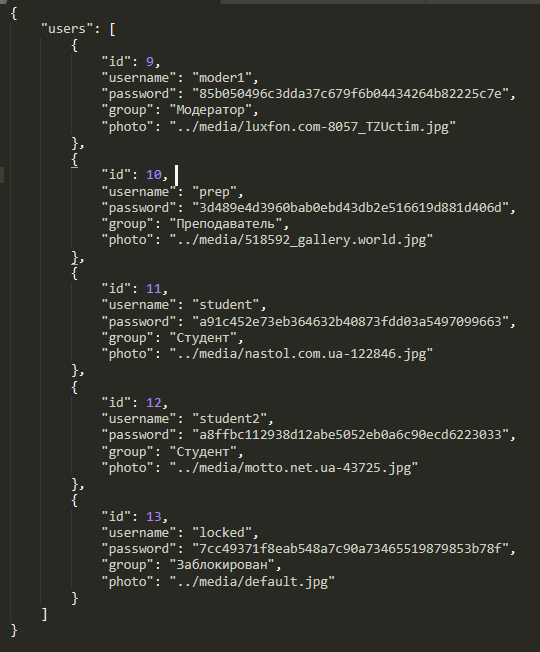


Рисунок 7 - users.json

3) Структура JSON файла с кафедрами - cafedras.json:

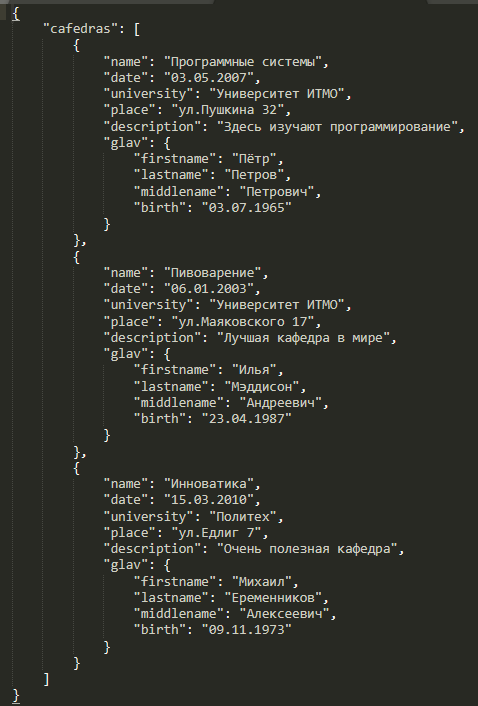


Рисунок 8 - cafedras.json

**5. Описание программного кода системы**

Все функции и методы, описанные ниже, реализованы в файлах /main/models.py и /main/views.py (считая от корневой папки проекта).

Изначально следует отметить, что все шаблоны проекта наследуются от шаблона index.html, в котором задана структура для каждой страницы, а также прописаны стили (использовался набор инструментов для стилизации Bootstrap).

При запуске системы пользователь в первую очередь попадает на страницу авторизации, с которой невозможно перейти на другую страницу сайта (пока пользователь не авторизуется). Данная страница формируется из шаблона auth.html, который наследуется от index.html (при этом ссылки в хедере не отображаются). В случае, если пользователь введет неверный пароль и/или логин система оповестит его сообщением. Если пользователь заблокирован модератором (имеет группу безопасности - "Заблокирован"), то он не сможет войти и система оповестит его о том, что он заблокирован. В случае если пользователь введет корректные логин и пароль, то он войдет в систему и попадет на главную страницу системы, которая формируется из шаблона home.html (на этой странице отображаются кафедры). Ниже представлены скриншоты этих страниц:

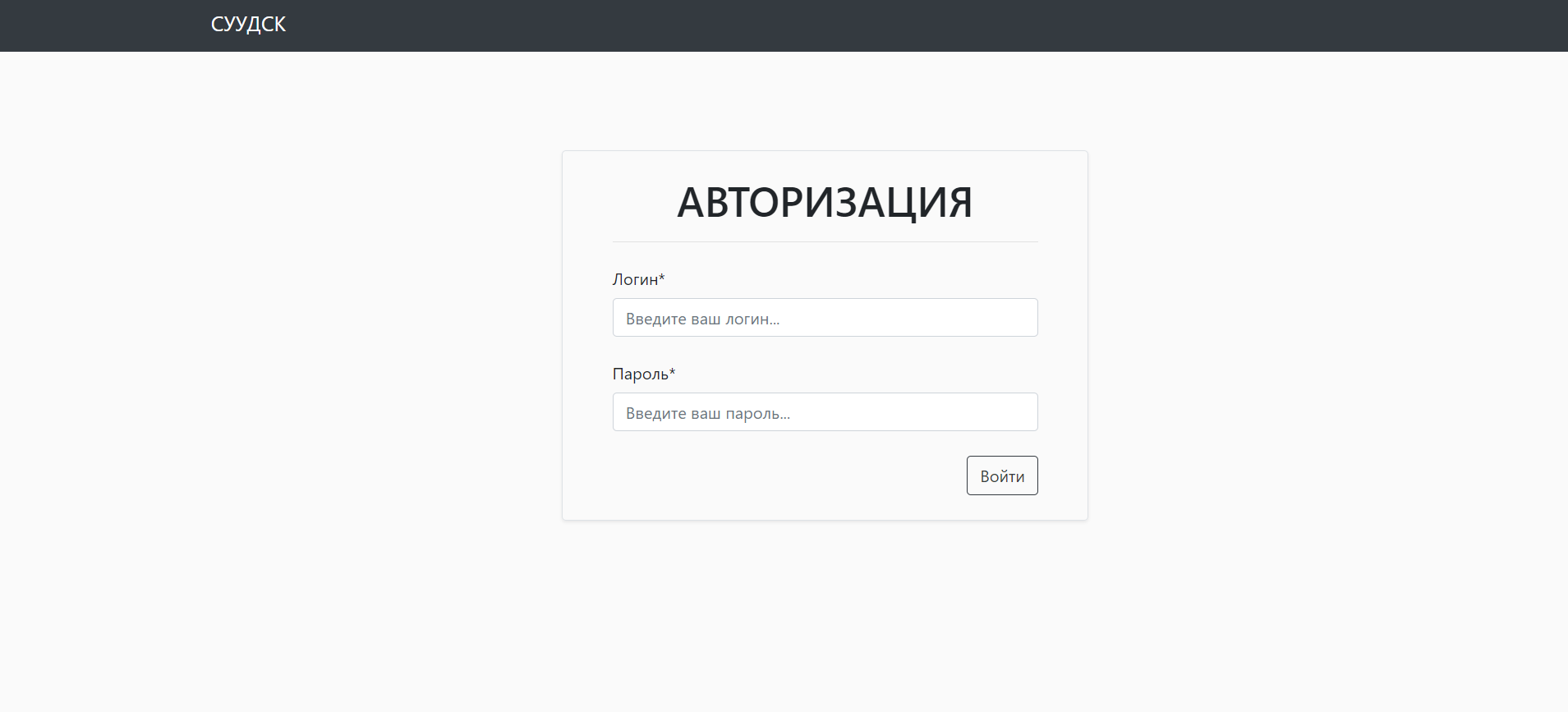


Рисунок 9 - Страница авторизации

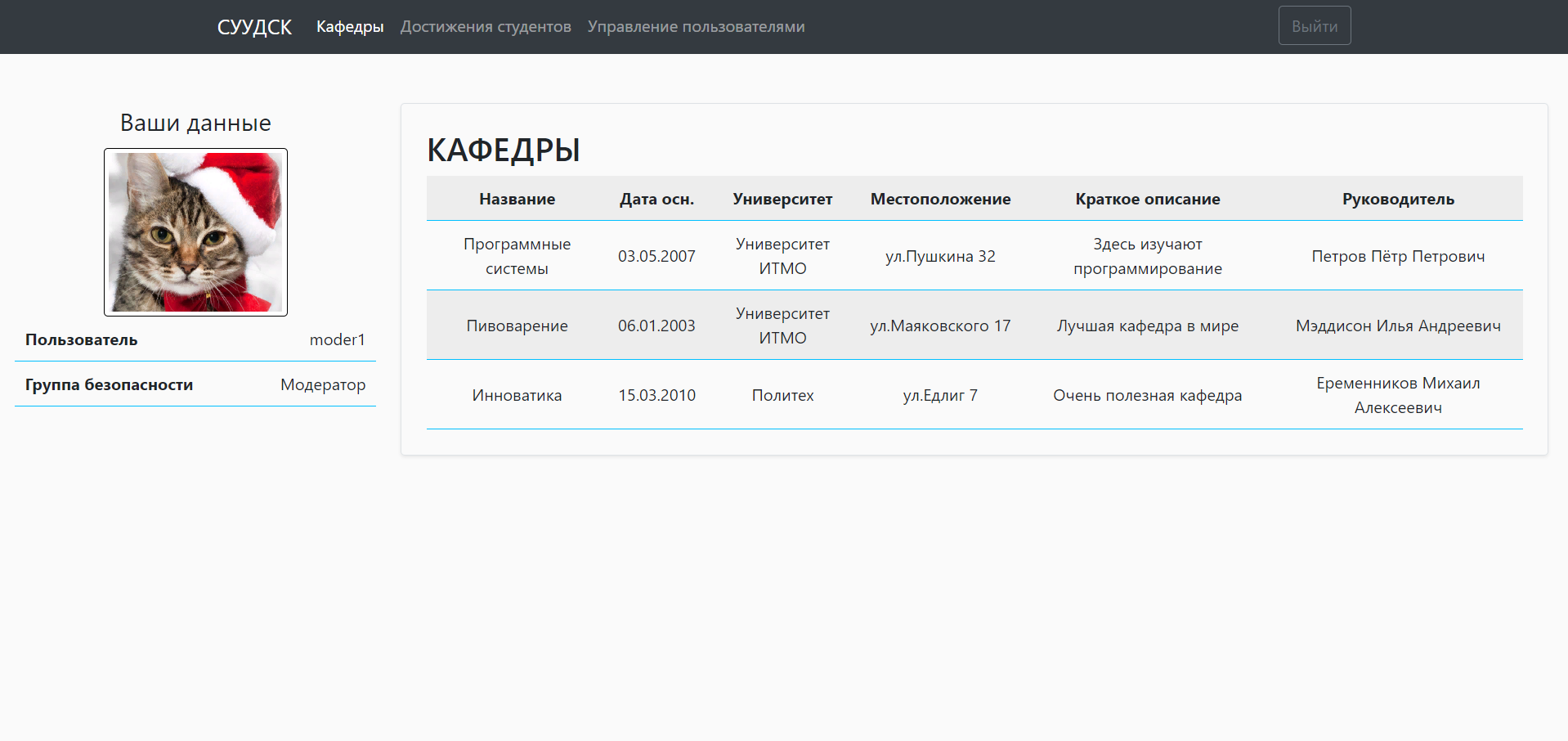


Рисунок 10 - Главная страница

Если пользователь является модератором, то в хедере отобразится ссылка "Управление пользователями", которая ведёт на одноимённую страницу. Данная страница формируется из шаблона manage.html. На данной страницы отображается список всех пользователей системы, а также действия, которые может с ними совершить модератор (изменение группы безопасности, удаление, блокировка), а также кнопка добавления нового пользователя, при нажатии на которую откроется окно добавления нового пользователя. Ниже показаны скриншоты этой страницы:

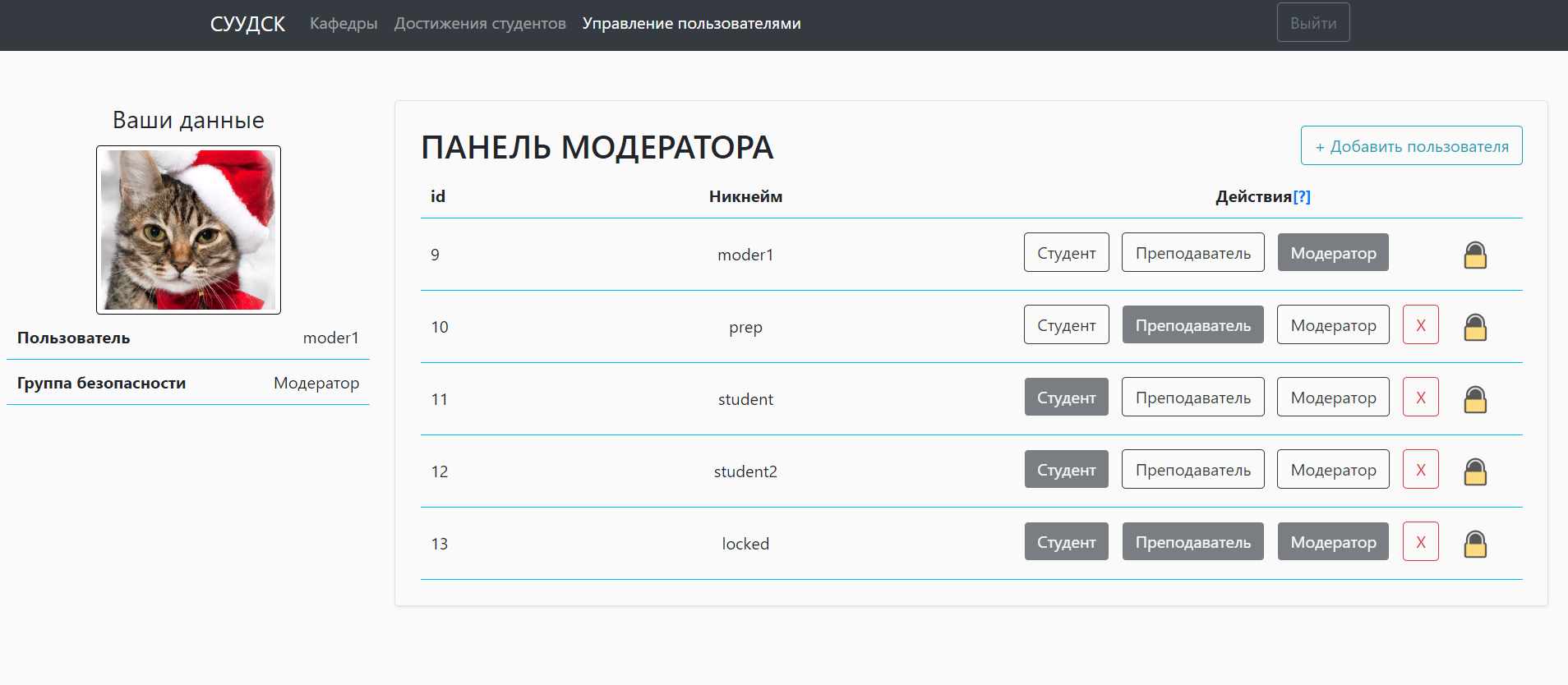


Рисунок 11 - Страница управления пользователями

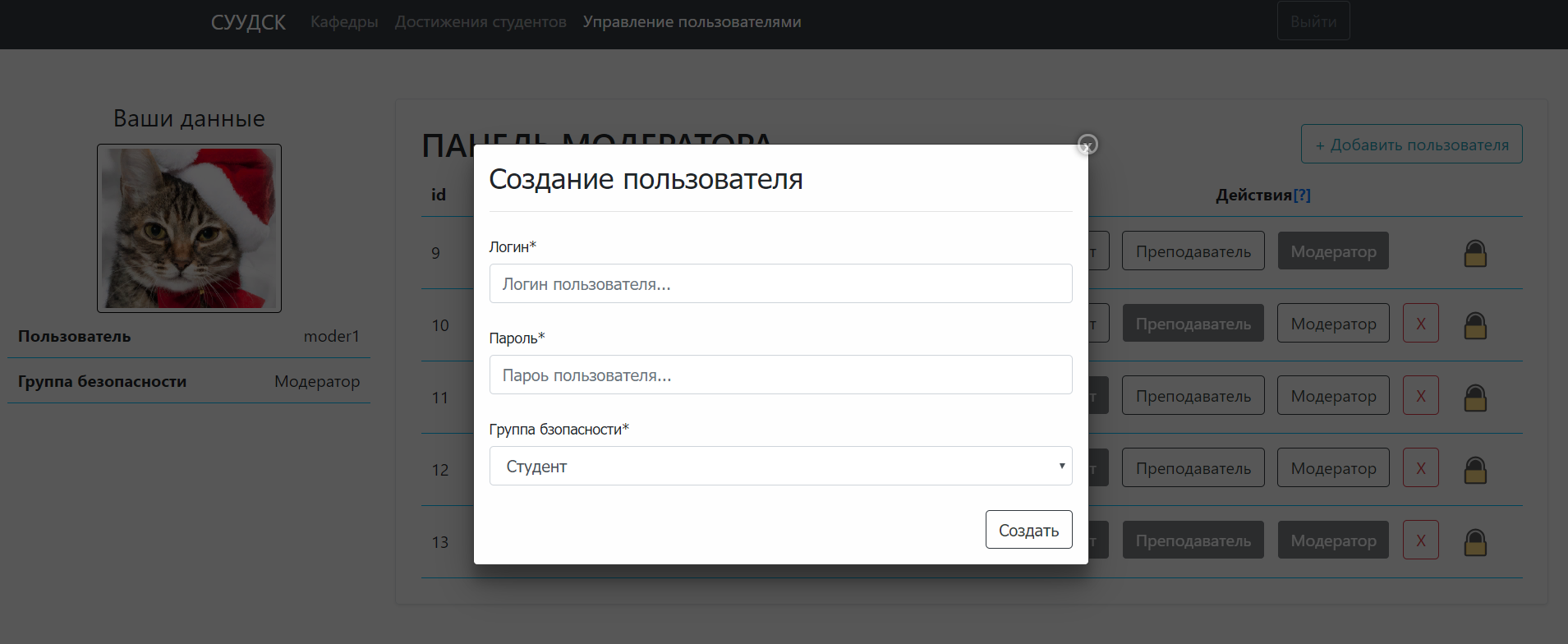


Рисунок 12 - Окно добавления пользователя

Страница "Достижения студентов" формируется из шаблона achiev.html и отображает достижения всех студентов. Отображение данной страницы зависит от группы безопасности пользователя, т.к. у каждой группы разный функционал:

* Если пользователь - "Студент", то ему будет доступна кнопка "Добавить достижение" и он может только просматривать свои и чужие достижения. Если пользователь попробует добавить от своего имени два достижения, одинаковых по наименованию, то второе достижение не будет добавлено и система оповестит пользователя, что он уже добавлял такое достижение. Если же пользователь добавляет достижение впервые, то система оповестит его об успешном добавлении достижения:

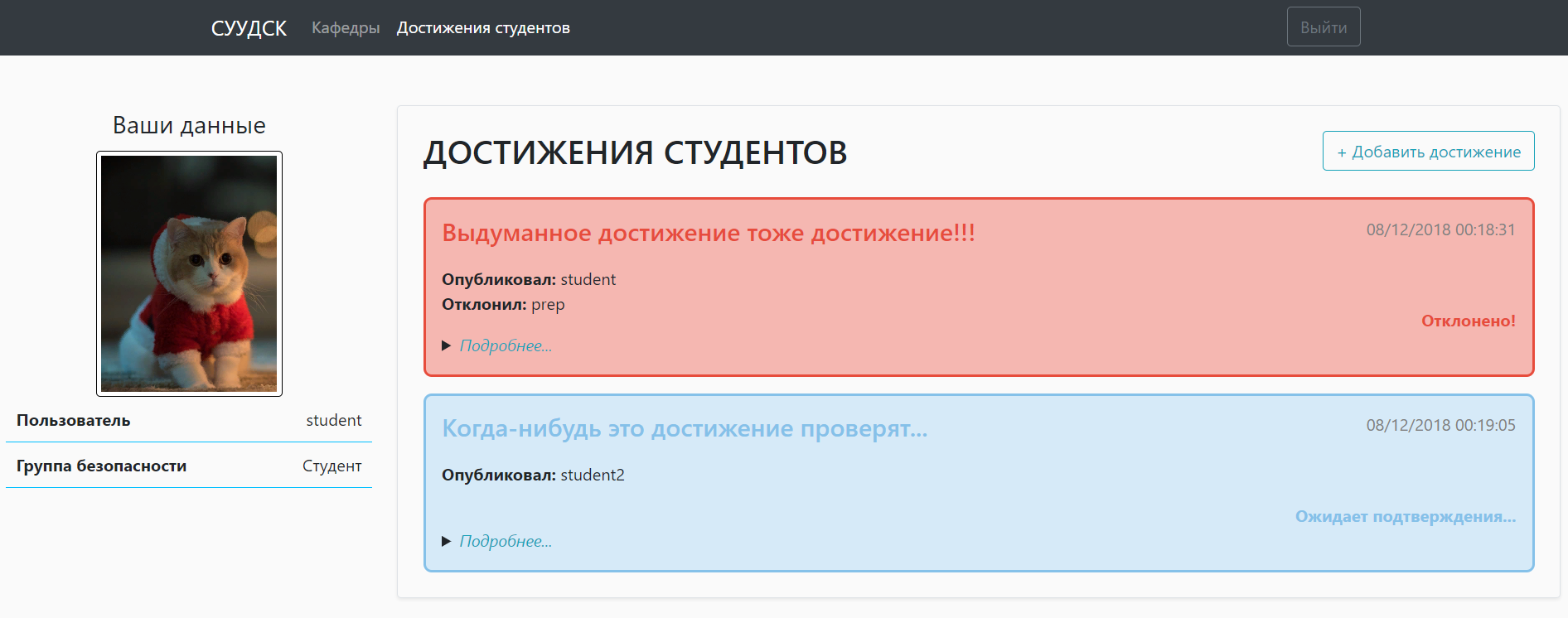


Рисунок 13 - Достижение студентов глазами студента

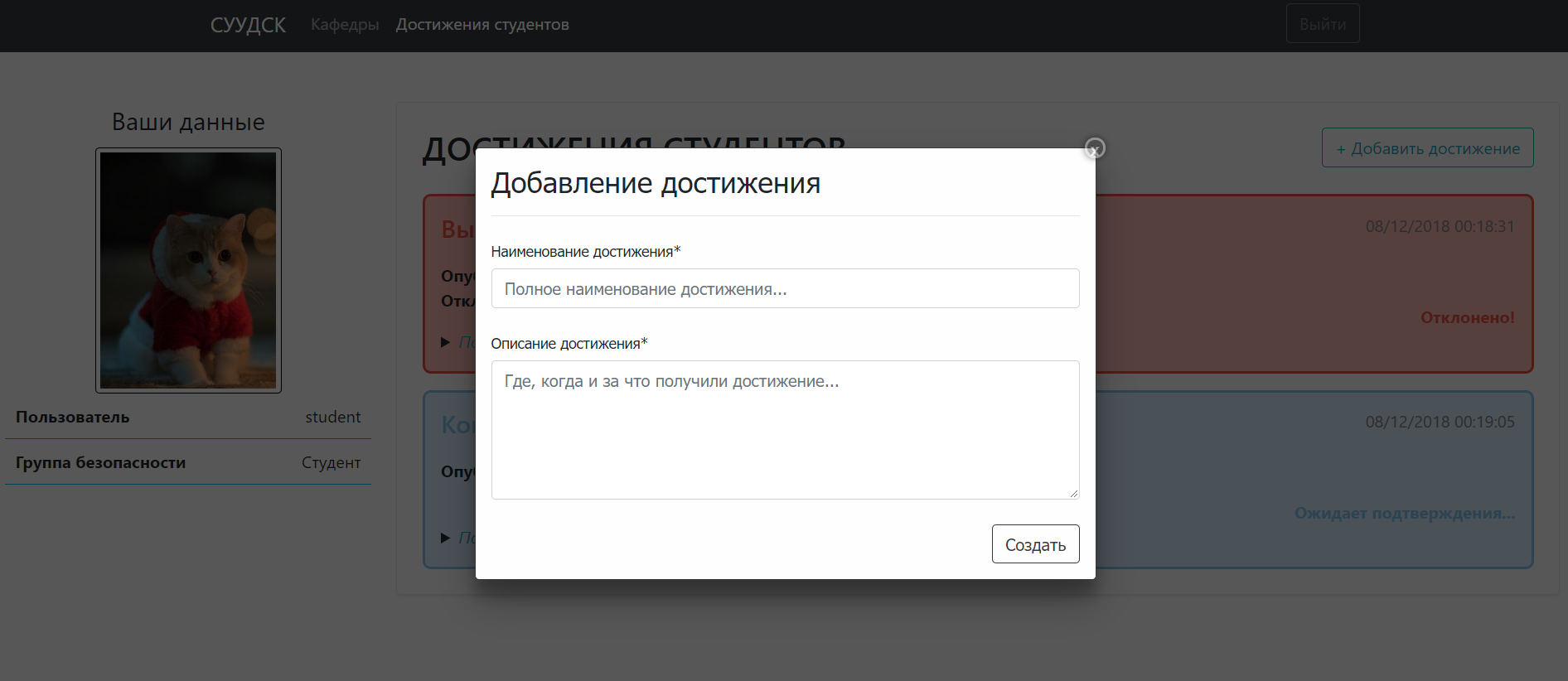


Рисунок 14 - Добавить достижение

* Если пользователь - "Преподаватель", то ему будет недоступна кнопка "Добавить достижения", но при этом на достижениях студентов, которые помечены как "Ожидает подтверждения…" (выделено синим цветом), у них будет отображены 2 кнопки - одобрить и отклонить достижение (при нажатии на одну из кнопок будет изменяться и достижение студента. Если одобрить достижение, то оно выделится красным цветом и пометится текстом "Одобрено!", если отклонить, то оно выделится красным цветом и пометится текстом "Отклонено!"). Ниже представлен скриншот страницы от лица преподавателя:

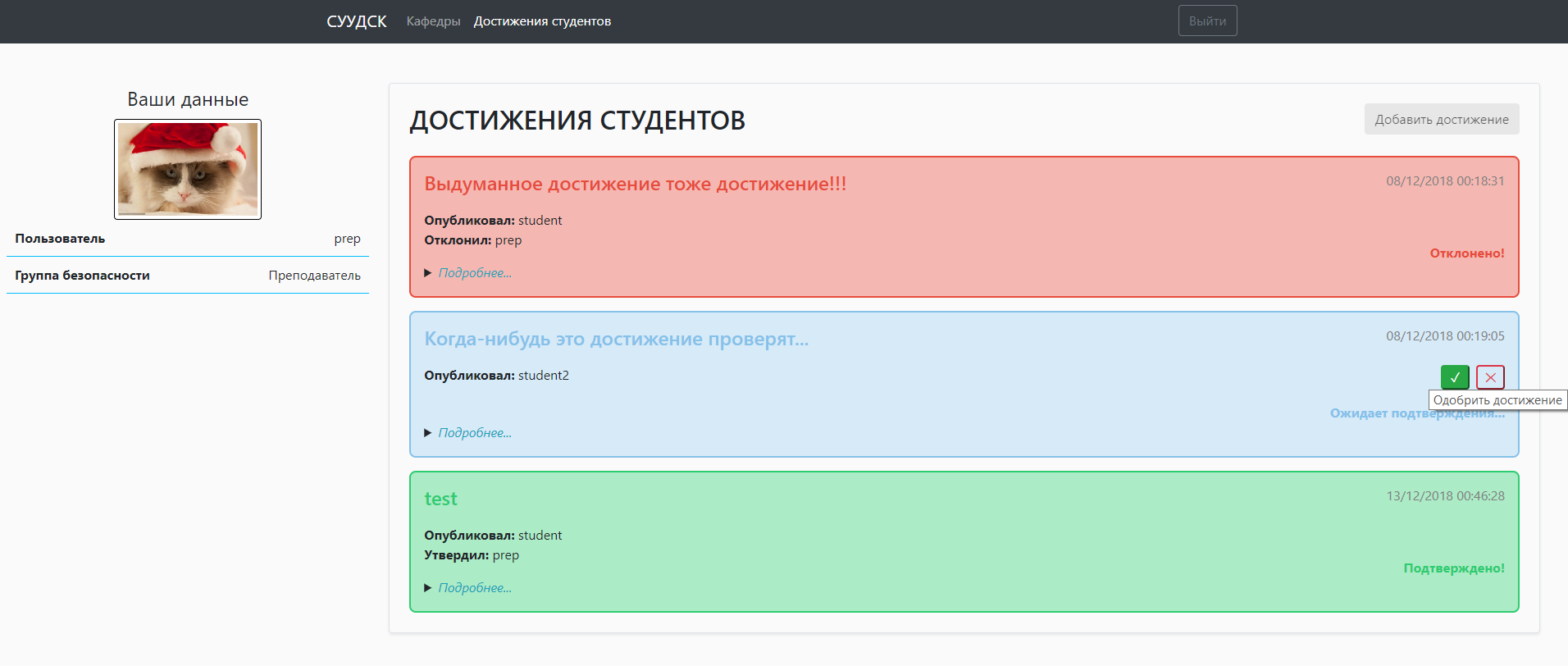


Рисунок 15 - Достижения студентов от лица преподавателя

* Если пользователь - "Модератор", то ему будет недоступна кнопка "Добавить достижение", но он сможет удалить любое достижение любого студента:

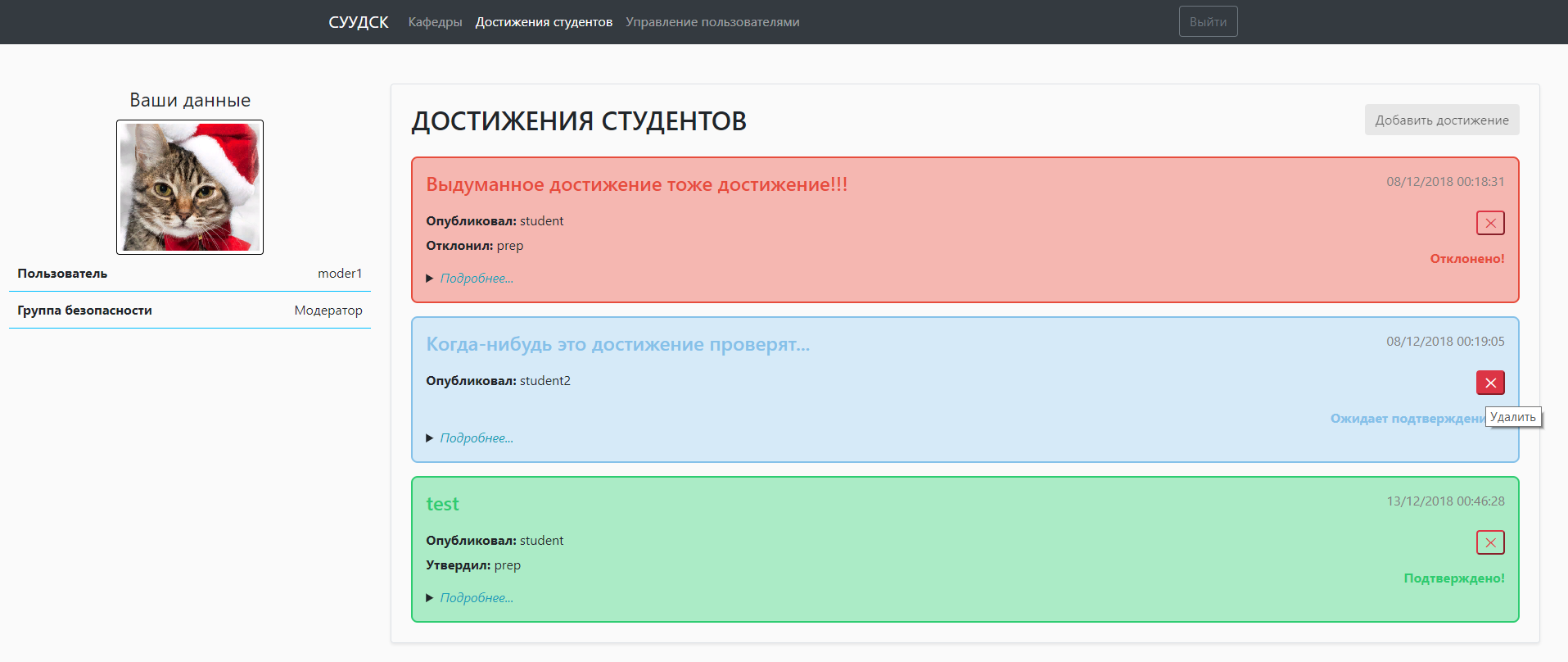


Рисунок 16 - Достижения студентов глазами модератора

Также с любой страницы при нажатии на фотографию откроется окно, в котором пользователь может сменить свою фотографию:

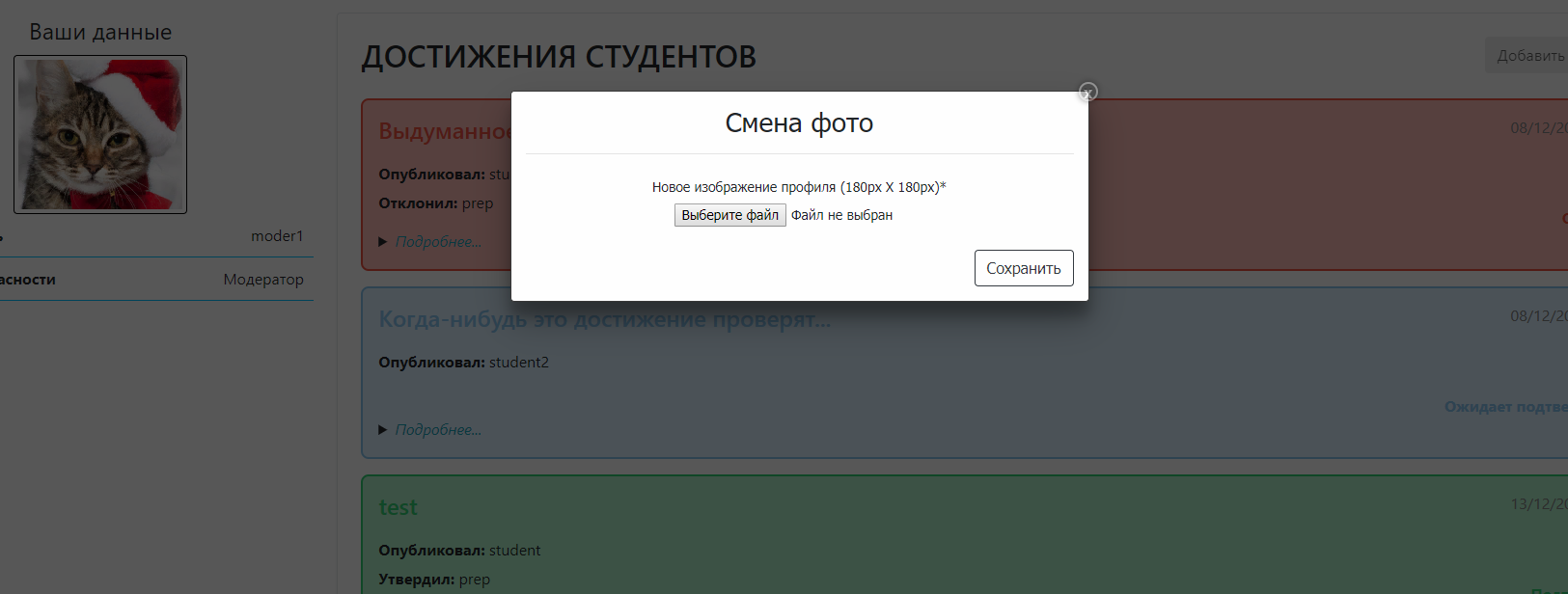


Рисунок 17 - Смена фото

Также в хедере страницы указана кнопка "Выйти" при нажатии на которую пользователь выйдет из своего профиля и окажется на странице авторизации.

Со всем функционалом системы можно ознакомиться, если запустить сервер с помощью файла "manage.py" (Если вы запускаете данный сервер с вашего компьютера впервые, то перед этим необходимо прописать в консоли две команды: "pip install Django-crispy-forms" и "pip install pillow" (если у вас не установлен pip, то перед этими командами необходимо его установить). Данные библиотеки используются только для стилизации пользовательских полей ввода). Ниже приведена таблица тестирования проекта (*таблица 3)*:

*Таблица 3. Тестирование проекта*

|  |  |  |
| --- | --- | --- |
| Тест | Ожидаемый результат | Фактический результат |
| Открытие неавторизованным пользователем какой-либо страницы сайта через адресную строку. | Пользователь останется на странице авторизации или выведется сообщение об ошибке. | Пользователь останется на странице авторизации. |
| Открытие авторизованным пользователем, не являющимся модератором, страницы с управлением пользователями. | Пользователь останется на странице или выведется сообщение об ошибке. | Пользователя переведёт на Главную страницу. |
| Открытие авторизованным пользователем страницы авторизации через адресную строку. | Пользователь останется на странице или выведется сообщение об ошибке. | Пользователя переведёт на Главную страницу. |
| Открытие пользователем, не являющимся студентом, окна добавления достижения. | Ничего не произойдет. | Ничего не произошло. |
| Ввод неверных данных в форму аутентификации. | Пользователь не сможет войти в систему. | Пользователь не смог войти в систему и система отобразила сообщение о некорректности введенных логина и/или пароля. |

*Таблица 3*

|  |  |  |
| --- | --- | --- |
| Одобрение/отклонение преподавателем достижения студента. | Корректное изменение json-файла. | Корректное изменение json-файла и динамическое отображение изменений. |
| Добавление модератором пользователя с существующим никнеймом. | Действие не выполнится. | Действие не выполнилось и вывелось сообщение о некорректном действии. |
| Добавление студентом достижений с одинаковыми наименованиями. | Первое достижение добавится, а остальные нет. | Первое достижение добавилось, а при достижении других система выводила сообщение о некорректном действии. |
| Смена пользователем фотографии. | Корректное изменение json-файла. | Корректное изменение json-файла и динамическое отображение изменений. |
| Выход пользователя с аккаунта. | Выход пользователя и переход на страницу авторизации. | Выход пользователя и переход на страницу авторизации. |
| Попытка входа заблокированного пользователя на сайт. | Пользователь не сможет войти в систему. | Пользователь не смог войти в систему и система отобразила сообщение о том, что он заблокирован в системе. |
| Добавление/удаление достижения студентом. | Корректное изменение json-файла. | Корректное изменение json-файла и динамическое отображение изменений, а также отображение системой сообщения о совершённом действии. |
| Добавление/изменение/удаление пользователя модератором. | Корректное изменение json-файла. | Корректное изменение json-файла и динамическое отображение изменений, а также отображение системой сообщения о совершённом действии. |

**Заключение**

По итогу курсовой работы была создана и спроектирована система учёта учебных достижений студентов кафедры. При этом на сайте была реализована функция добавления фотографий пользователями, а также было реализовано шифрование паролей пользователями (пароль шифруется при создании нового пользователя, в середину пароля вставляется никнейм пользователя, после чего вся эта строка шифруется методом sha1. В итоге получается примерно такая строка: ' 85b050496c3dda37c679f6b04434264b82225c7e'). Также дополнительно была реализована работа с сессиями, аутентификация, использование инструментария стилизации Bootstrap.

По итогу выполнения работы было получено следующее: 1 Django проект (178 строк в views.py, 136 строк в models.py, 24 строки в forms.py), 3 json файла (считать кол-во строк не имеет смысла, т.к. эти файлы динамически изменяются и при новом изменении все символы помещаются в одну строку), 5 шаблонов html (409 строк), 1 пояснительная записка (22 страницы), 3 таблицы, 17 рисунков, 1 презентация.

**Список используемых источников**

1. Лямин А.В., Череповская Е.Н. Объектно-ориентированное программирование. Компьютерный практикум. – СПБ: Университет ИТМО, 2017. – 143 с.

2. Инструментарий Bootstrap - [https://getbootstrap.com]

3. Библиотека для изменения пользовательских полей ввода данных django-crispy-forms - [https://github.com/django-crispy-forms/django-crispy-forms]