МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Воронежский государственный университет»

Факультет компьютерных наук

Кафедра информационных систем

09.03.02 Информационные системы и технологии

**Отчёт об учебной практике**

*Веб-проект «Система для выставления оценок»*

Обучающийся студентка 3 курса А.Д. Ильина

Руководитель доц. А.В. Сычёв

Воронеж 2018

Оглавление

[Постановка задачи 2](#__RefHeading___Toc170_3860881691)

[Метод решения задачи 3](#__RefHeading___Toc172_3860881691)

[Инструментальные средства, использованные для разработки приложения 4](#__RefHeading___Toc174_3860881691)

[Описание функционала приложения и интерфейса пользователя 5](#__RefHeading___Toc176_3860881691)

[Схема данных и описание структуры разработанного приложения 8](#__RefHeading___Toc178_3860881691)

[Оценка степени завершённости и перспективы доработки приложения 10](#__RefHeading___Toc180_3860881691)

[Ссылки 11](#__RefHeading___Toc647_1887565634)

## Постановка задачи

Необходимо разработать приложение для формирования таблиц с аттестационными и экзаменационными оценками студентов. Приложение должно обладать следующим функционалом:

* создание таблиц с информацией о студентах и предметах с возможностью их редактирования;
* формирование таблиц с оценками студентов по предметам, внесенным в приложение;
* авторизация пользователей с различными уровнями доступа: администраторы (могут вносить изменения во все таблицы) и студенты (могут просматривать только свои оценки и список предметов).

## Метод решения задачи

Задача решена при помощи микро-фреймворка Flask языка Python. Приложение содержит клиентскую часть, серверную часть и базу данных. База данных приложения содержит все данные о предметах, студентах и оценках, используемые в программе. В серевной части приложения применяется архитектура MVC (model-view-controller) для разделения модуля работы с базой данных, модуля представления и бизнес-логики. Также с помощью технологий AJAX и JSON выполняются асинхронные запросы на сервер от клиента для вывода и редактирования таблиц.

## Инструментальные средства, использованные для разработки приложения

При разработке серверной части приложения использовались следующие средства:

* Python 3.7
* библиотеки Python: Flask, Flask-Login, Flask-SQLAlchemy\*

При разработке клиентской части приложения использовались следующие средства:

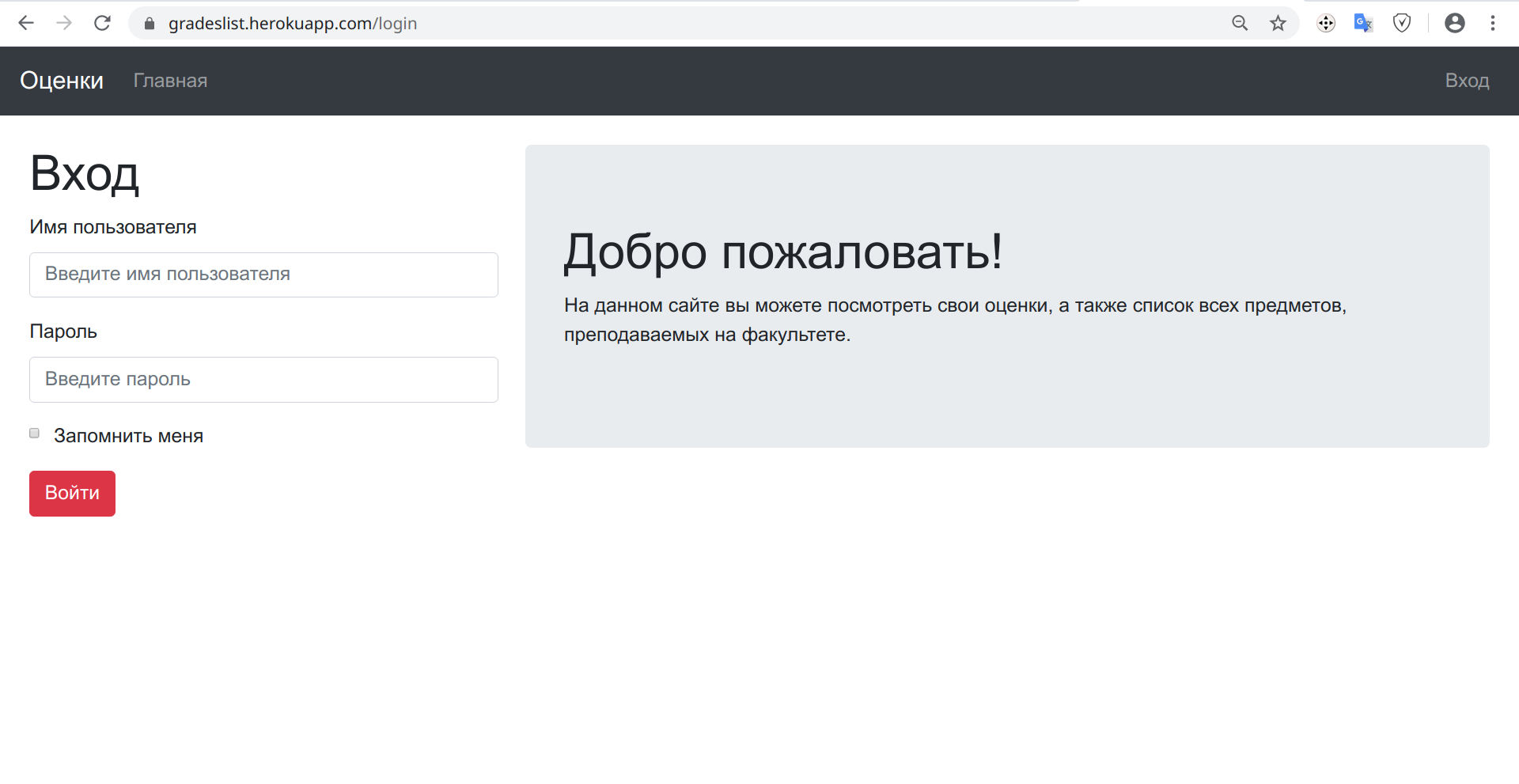
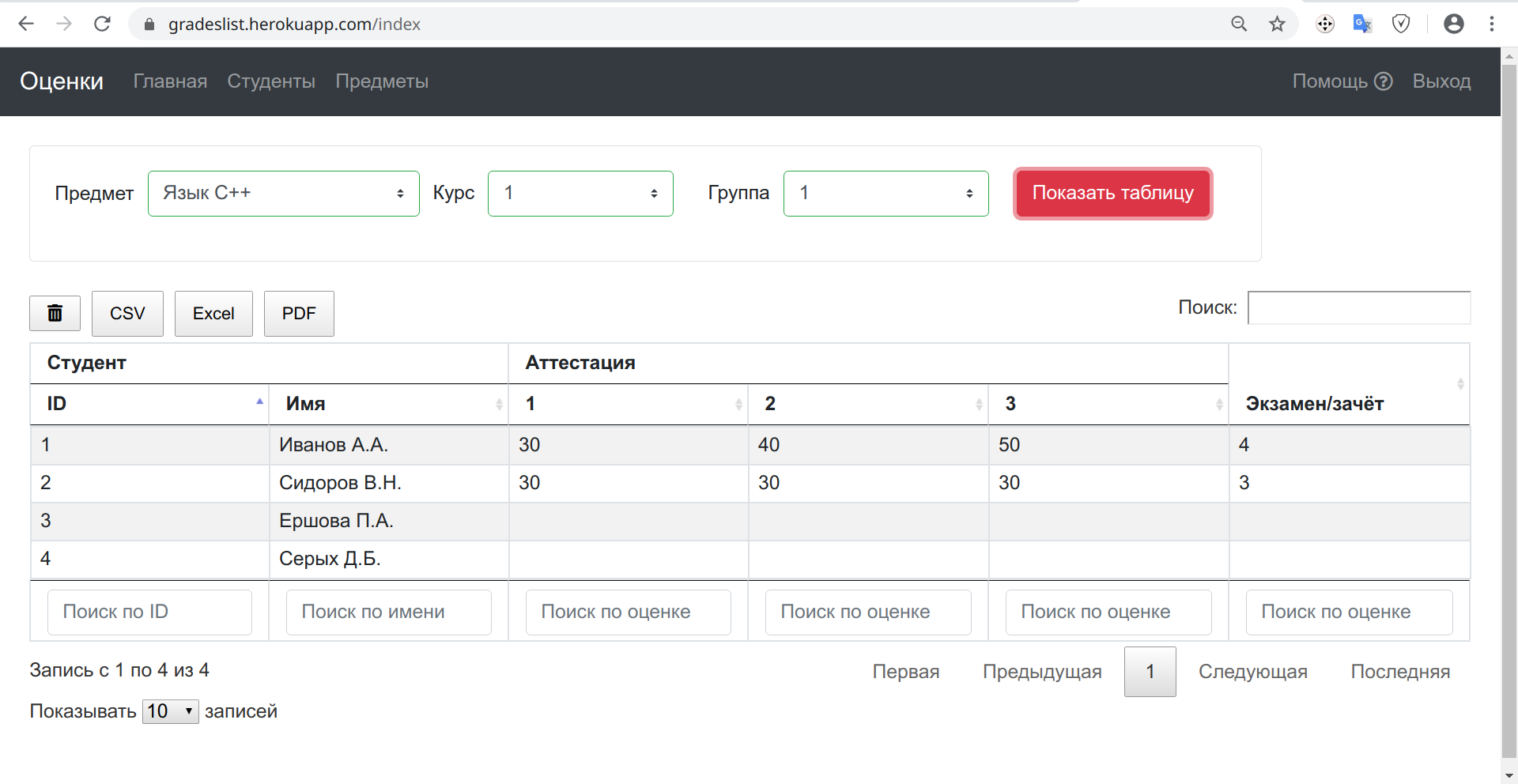
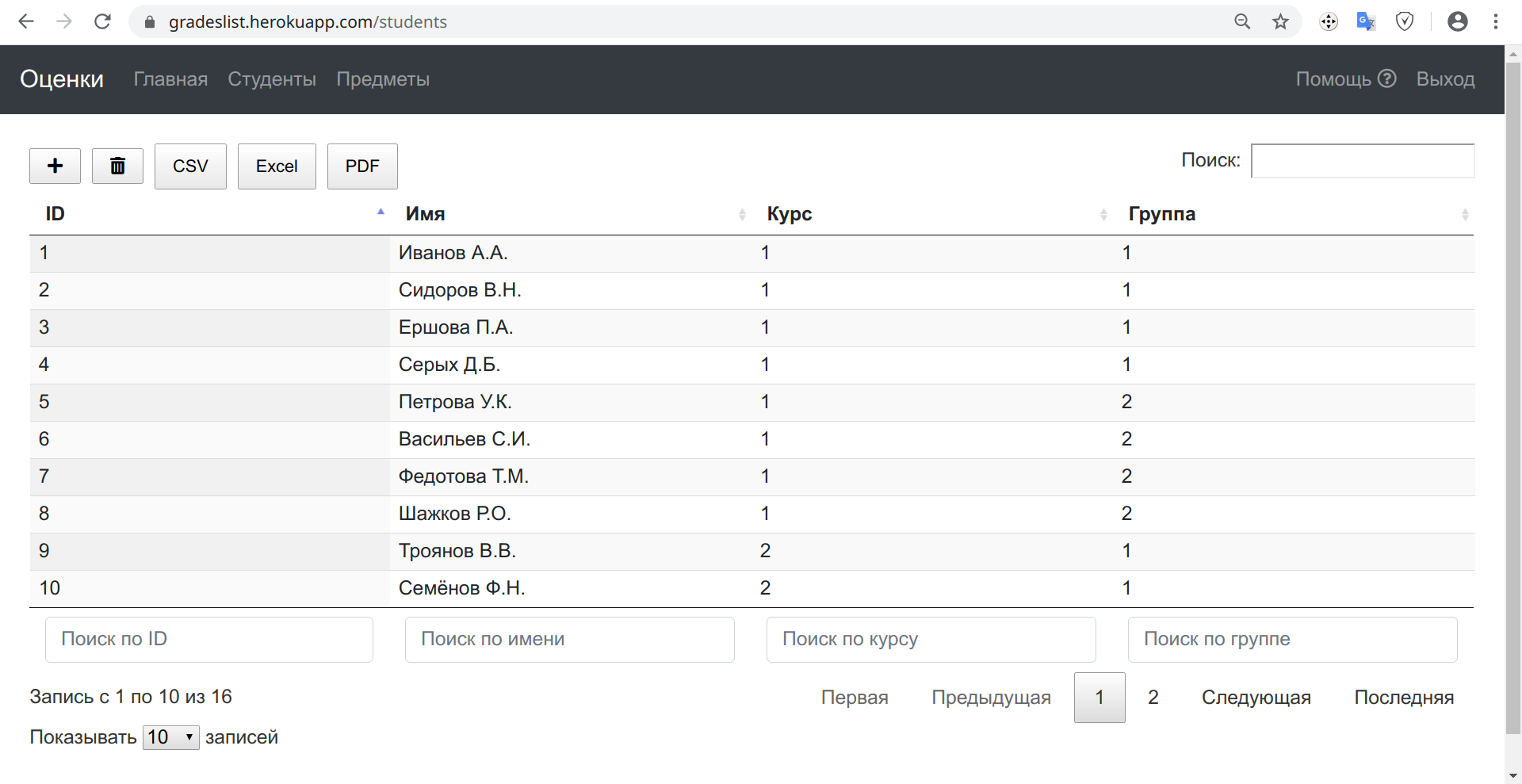
* HTML
* CSS
* JavaScript и библиотеки JavaScript: jquery, DataTables, Bootstrap
* AJAX

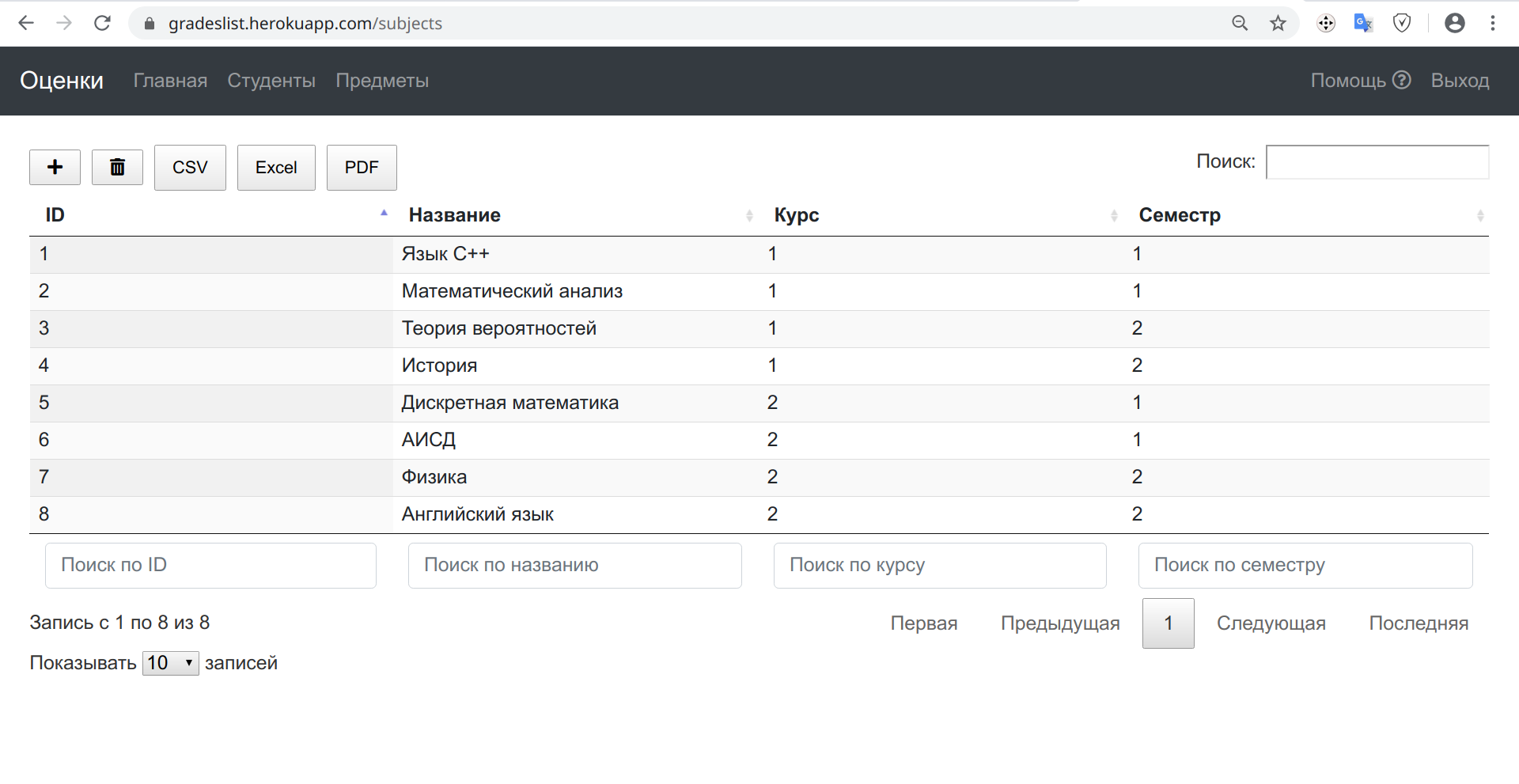
Приложение развёрнуто на хостинге Heroku. В качестве базы данных используется PostgreSQL, т.к. она бесплатно предоставляется хостингом.

\* Полный список использованных библиотек см. в файле gradeslist/requirements.txt

## Описание функционала приложения и интерфейса пользователя

Приложение обладает функционалом, описанным в постановке задачи. На всех страницах расположена панель навигации. Приложение содержит следующие страницы:

* /login — страница авторизации. Содержит форму пользователя с двумя полями ввода (имя пользователя и пароль) и кнопку подтверждения
* /index — начальная страница. На ней находится таблица с оценками. Для пользователя-студента таблица выводит все его оценки за текущий семестр. Для администратора предлагается выбрать в форме предмет, курс и группу, после чего при нажатии на кнопку «Показать таблицу» выводится таблица для данных предмета и группы
* /students — таблица со списком студентов, доступна только администратору
* /subjects — таблица предметов. Администратор может редактировать таблицу, студенты — только просматривать

Все таблицы в приложении являются редактируемыми для администратора и имеют следующие функции:

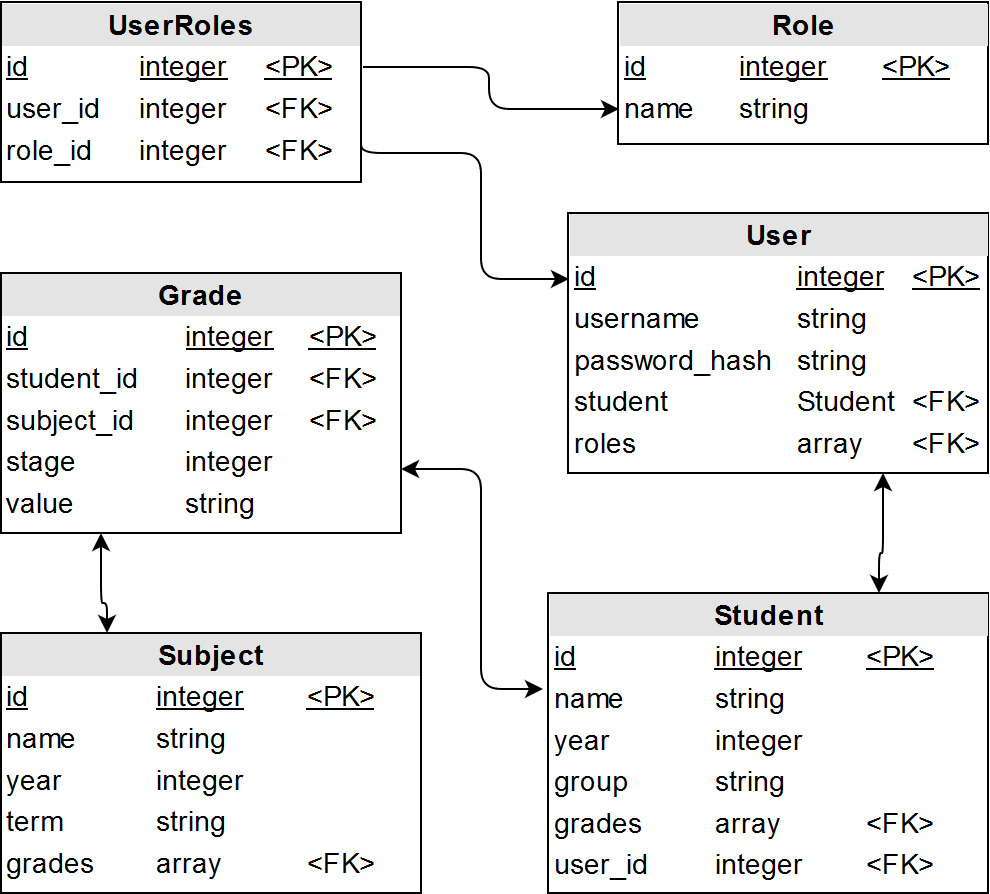
* добавление рядов (для таблиц студентов и предметов)
* редактирование ячеек (для таблицы оценок) и рядов (для остальных таблиц) при двойном нажатии на ячейку/ряд
* выбор ячеек/рядов при нажатии мышкой и их удаление
* экспорт таблиц в форматы CSV, Excel и PDF
* постраничный вывод
* фильтрование полей
* сортировка полей

## Схема данных и описание структуры разработанного приложения

База данных приложения содержит следующие таблицы:

* Пользователи (класс User)
* Роли пользователей (класс Role)
* Предметы (класс Subject)
* Студенты (класс Student)
* Оценки (класс Grade)

Схема данных:



Структура приложения:

**gradeslist\**  // Корневая папка

db/

*app.db* // Локальная база данных

logs\ // Лог-файлы

migrations\ // Файлы миграции БД

static\

css\ // Стили css

js\ // Скрипты javascript

templates\ // Шаблоны страниц

*models.py* // Модели базы данных

*routes.py* // Контроллеры

*config.py*  // Файл конфигурации

*Procfile* // Файл Heroku

*requirements.txt*  // Требования к окружению

*run.py*  // Запуск приложения

Приложение разделено на несколько py-модулей. Запуск программы осуществляется в скрипте *run.py*. Все файлы клиентской части приложения (HTML, JS) содержатся в папке static.

## Оценка степени завершённости и перспективы доработки приложения

В настоящий момент приложение может полностью функционировать и выполнять поставленные задачи. Тем не менее, можно внести некоторые доработки для удобства пользователя, таких как проверку полей таблиц (например, на наличие одинаковых записей), возможность выводить оценки всего курса в одной таблице и т.п. Также необходимо улучшить авторизацию студентов (возможно, осуществление входа на сайт по номеру студенческого билета и дате рождения).

Приложение имеет потенциал к расширению. Осуществимые доработки включают в себя:

* добавление пользователей-старост (могут просматривать оценки своей группы) и пользователей-преподавателей (могут изменять оценки по своим предметам)
* таблица с расписанием предметов в текущем семестре
* таблицы посещаемости занятии и т.д.

## Ссылки

1. Ссылка на развёрнутое приложение: <https://gradeslist.herokuapp.com/login>

ВАЖНО! Приложению может потребоваться до 30с для того, чтобы загрузиться. Логин и пароль для входа администратора; *admin*. Логин и пароль для входа студента; *1*.

1. Ссылка на репозиторий в gitHub: <https://github.com/ussur/gradeslist>