

Лабораторная работа №16

Django

1 Цель работы

Изучить Django и научиться применять полученные знания на практике.

2 Краткая теория

Перед тем, как приступить к заданиям можно почитать руководство по созданию веб-приложений с помощью фреймворка Django <https://metanit.com/python/django/>, также можно почитать книгу, если, конечно, интересно, если нет, то нет.

Постолиг А. В.

Python, Django и PyCharm для начинающих. -СПб.: БХВ-Петербург, 2021 -464 с.: ил. -(Для начинающих)

3 Задания для выполнения работы

Задание 1.

Задачи.

1. Выбрать какие-нибудь 3 таблицы из предыдущей лабораторной работы (напомню, что их должно было быть не менее 5).

2* (НЕОБЯЗАТЕЛЬНО). Взять все таблицы.

Задание 2.

На основе приведенных примеров создать по 3 формы для каждой из таблиц, выбранных в задании 1: 1 форма для ввода записей в таблицу, 1 для вывода ВСЕХ записей, 1 для обновления КАКОЙ-НИБУДЬ КОНКРЕТНОЙ записи.

Задание 3.

Предусмотреть возможность удаления КАКОЙ-НИБУДЬ КОНКРЕТНОЙ записи.

Задание 4* (НЕОБЯЗАТЕЛЬНО).

Если какое-либо поле формы заполнено неправильно, стоит немедленно сообщить об этом, а не ждать, пока форма будет отправлена. Если форма заполнена корректно, то отправить ее и сообщить, что форма была отправлена успешно.

Задание 5.

Задачи.

1. ОБЯЗАТЕЛЬНО предусмотреть возможность авторизации.
2. ОБЯЗАТЕЛЬНО предусмотреть возможность регистрации.
3. Для каждого пользователя НУЖНО хранить как минимум следующие сведения: фамилия, имя, отчество, e-mail и пароль.

4. Фамилия, имя и отчество ДОЛЖНЫ начинаться с прописной буквы, все остальные буквы ДОЛЖНЫ быть строчными. Фамилия, имя и отчество ДОЛЖНЫ содержать буквы на кириллице. Проверку фамилии, имени, отчества на соответствие указанным критериям реализовать с помощью регулярного выражения.

Про регулярные выражения можно почитать здесь <https://pythonist.ru/regulyarnye-vyrazheniya-v-python/>.

5* (НЕОБЯЗАТЕЛЬНО). E-mail должна быть вида **логин@домен**, где домен может быть второго, третьего, четвертого и пятого уровней. E-mail ДОЛЖНА содержать латинские буквы. Проверку e-mail на соответствие указанным критериям реализовать с помощью регулярного выражения.

6* (НЕОБЯЗАТЕЛЬНО). Длина пароля должна быть не менее 8 символов. Пароль должен содержать прописные и строчные латинские буквы, цифры, специальные символы. Проверку пароля на соответствие указанным критериям реализовать с помощью регулярного выражения.

7. Пароль ОБЯЗАТЕЛЬНО хранить в хешированном виде. Хешировать можно, например, с помощью алгоритма MD5 или алгоритма SHA256.

Про данные алгоритмы можно почитать здесь <https://python-scripts.com/md5-sha1>.

8* (НЕОБЯЗАТЕЛЬНО). Написать собственный метод, который осуществляет хеширование, выполняемое алгоритмом MD5 или алгоритмом SHA256.

Про алгоритм MD5 можно почитать здесь <https://ru.wikipedia.org/wiki/MD5>, про алгоритм SHA256 можно почитать здесь <https://ru.wikipedia.org/wiki/SHA-2>.

9* (НЕОБЯЗАТЕЛЬНО). Так как пароль может быть утерян, нужно предусмотреть возможность восстановления пароля по e-mail.

Про то, как работать с почтой, можно почитать здесь <https://python-scripts.com/send-email-smtp-python>.

Задание 6.

1. Также ДОЛЖНЫ быть добавлены роли (как минимум администратор и обычный пользователь).

2. В системе ДОЛЖЕН быть хотя бы один администратор.

3* (НЕОБЯЗАТЕЛЬНО). По долгу службы администратор обязан иметь возможность выдавать роли обычным пользователям.

4. Администратор ОБЯЗАН иметь полный доступ ко всем объектам, в том числе созданными обычными пользователями.

5. Обычный пользователь ОБЯЗАН иметь доступ только к тем объектам, которые он создал.

Задание 7.

Для КАЖДОЙ из таблиц, выбранных в задании 1, реализовать выгрузку данных в Excel, Word и pdf. Можно в тех же формах, в которых выводятся все записи, можно в отдельных формах.

Про выгрузку данных в Excel можно почитать здесь <https://docs-python.ru/packages/modul-openpyxl/> и здесь <https://zetcode.com/python/openpyxl/>.

Про работу с файлами в Word можно почитать здесь <https://tokmakov.msk.ru/blog/item/78>.

Про работу с pdf-файлами можно почитать здесь <https://blog.aspose.com/ru/pdf/create-pdf-files-in-python/>.

Задание 8* (НЕОБЯЗАТЕЛЬНО).

Можно использовать Bootstrap для более красивого отображения страниц.

Задание 9* (НЕОБЯЗАТЕЛЬНО).

Можно использовать AJAX, чтобы лишний раз не перезагружать страницы.