

# Домашнее задание 4

## Авторизация и обработка форм

Целью домашнего задания является подключение встроенной системы авторизации Django и разработка основных форм добавления данных.

### 1. Авторизация и регистрация пользователей

Поскольку мы используем встроенную в Django систему пользователей, имеет смысл использовать готовые методы для логина и регистрации пользователей. Не забывайте, что при создании модели стандартного пользователя User вы должны так же создать связанный Profile, в котором хранятся дополнительные поля.

### 2. Необходимые формы авторизации

- Форма логина. Располагается по URL **/login/**. Пользователь логин/пароль. Дополнительно в GET параметрах передается параметр **continue** - URL на который нужно отправить пользователя после успешной авторизации. При неудачной авторизации нужно отображать сообщение об ошибке.
- Форма регистрации. Располагается по URL **/signup/**. Пользователь вводит все необходимые поля (email, имя, пароль). После успешной регистрации отправляется на главную страницу.
- Ссылка “выход”. Располагается на каждой странице в шапке сайта. Видна только авторизованным пользователям. После выхода пользователь остается на текущей странице.
- Форма редактирования профиля. Располагается по URL **/profile/edit**. Доступна только для авторизованного пользователя. Пользователь видит все необходимые поля: email, nick, avatar. После сохранения остается на странице.

### 3. Необходимые формы добавления данных

- Форма добавления вопроса. Располагается по URL **/ask/**. Пользователь вводит название, текст и теги вопроса. После успешного добавления отправляется на страницу вопроса.
- Форма добавления ответа. Располагается на странице вопроса URL **/question/**. Пользователь вводит только текст ответа. После успешного добавления необходимо отправить пользователя на нужную страницу ответов данного вопроса и проскролить страницу так, чтобы добавленный ответ был виден.

### 4. Требования предъявляемые к формам

Ко всем формам предъявляются следующие требования:

- Обязательна валидация входных данных и вывод сообщений об ошибках.
- При выводе сообщений об ошибках введенные пользователем данные сохраняются, так что нет необходимости вводить их повторно.
- Все формы отправляются с методом POST.
- После успешной обработки формы пользователь отправляется на новую страницу с помощью редиректа.
- Использовать встроенную в Django защиту от **CSRF**.
- Для обработки и вывода форм использовать **django.forms**.

## 5. Полезные ссылки

- Документация по [системе авторизации](#);
- [Как войти залогинить пользователя](#);
- [Как разлогинить пользователя](#);
- [Как зарегистрировать пользователя](#);
- Тutorial по [проектированию форм в Django](#).