

Модуль 4 Занятие 8

# Flask: выполнение заданий



**Маршруты**

**Шаблоны**

**Формы**

**ORM**

**CRUD**

**+**

**Новый проект  
«Укорачиватель ссылок»**

# Проект «Укорачиватель ссылок»

Ссылки на страницы или другие документы в интернете могут быть длинными, сложными и некрасивыми. Делиться такими ссылками не всегда удобно и поэтому существуют специальные сервисы, которые могут укорачивать ссылки.

Создадим свой собственный сервис — проект «Укорачиватель ссылок»!

Ссылка на это изображение:

[https://live.staticflickr.com/5167/5340309206\\_e592a0c4c3\\_b.jpg](https://live.staticflickr.com/5167/5340309206_e592a0c4c3_b.jpg)





# Функционал проекта

Проект должен состоять из двух страниц и одного эндпоинта:



**Главная страница '/'**. На главной выводится одно поле для вставки исходной ссылки и кнопка «Получить короткую ссылку». Для поля должна быть настроена валидация: поле не может быть пустым и может содержать только ссылку. В случае ошибок должен быть показан текст ошибки. Если валидация прошла успешно, то пользователь перенаправляется на страницу с короткими ссылками.

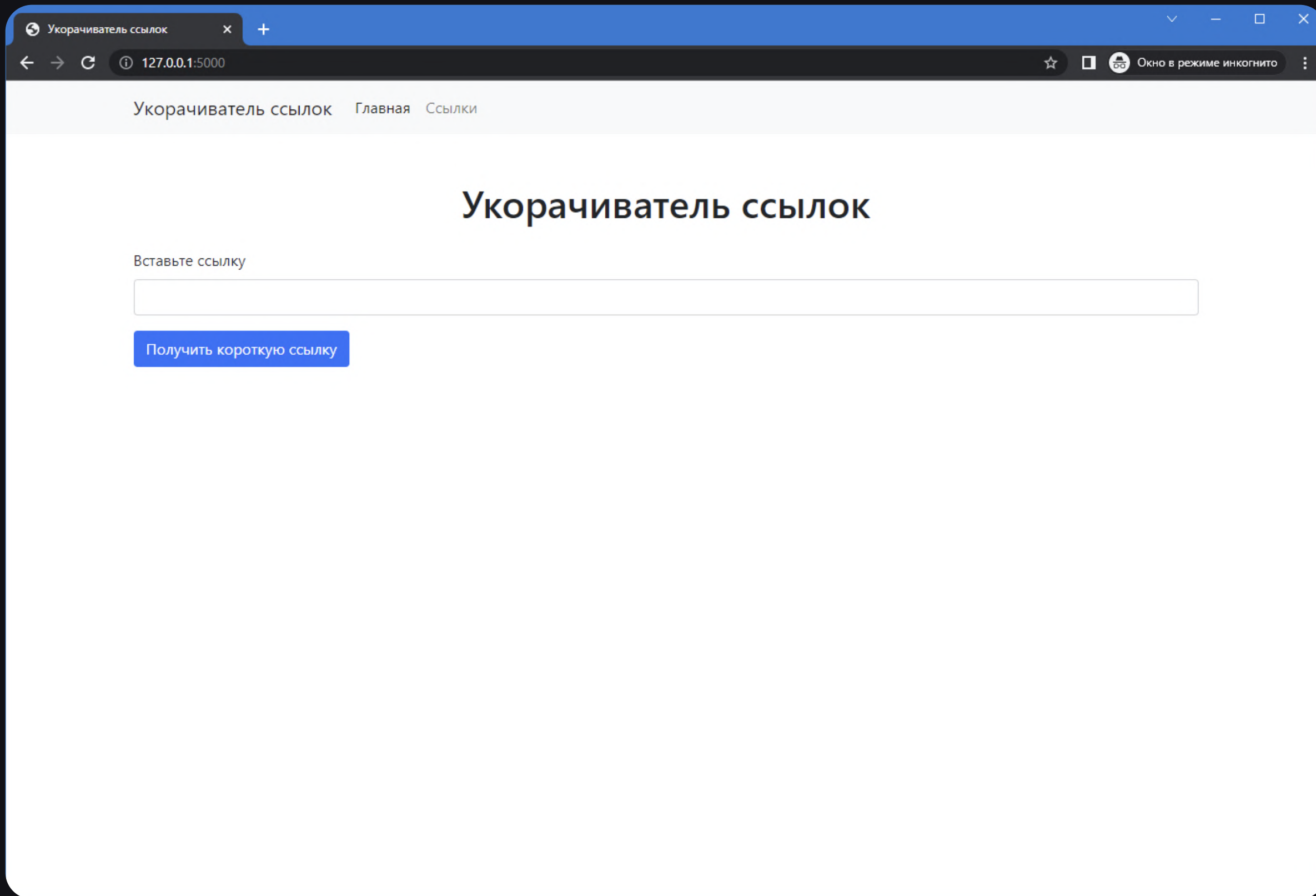


**Страница с созданными ссылками '/urls'**. На странице отображается таблица со столбцами: №, Исходная ссылка, Короткая ссылка, Дата создания, Количество переходов. Короткая ссылка имеет длину 6 символов и состоит из английских букв и цифр и формируется случайным образом. При этом она должна быть уникальной. Ссылки на странице располагаются в порядке от самых новых ссылок, к самым старым, столбец «Количество переходов» отображает сколько раз был выполнен переход по короткой ссылке.



**Эндпоинт '/<short>'**. При переходе по короткой ссылке, например <http://127.0.0.1:5000/Gprebb>, пользователь перенаправляется на страницу с исходной ссылкой, а для этой короткой ссылки в таблице увеличивается значение «Количество переходов».

# Главная страница



# Страница со ссылками

Укорачиватель ссылок

127.0.0.1:5000/urls

Окно в режиме инкогнито

Укорачиватель ссылок

Главная

Ссылки

## Укорачиватель ссылок

#	Исходная ссылка	Короткая ссылка	Дата создания	Количество переходов
1	<a href="https://cdn.jsdelivr.net/np...">https://cdn.jsdelivr.net/np...</a>	<a href="http://127.0.0.1:5000/ilWYJl">http://127.0.0.1:5000/ilWYJl</a>	04.042023 15:27:42	2
2	<a href="https://i.imgflip.com/44lr3r.jpg">https://i.imgflip.com/44lr3r.jpg</a>	<a href="http://127.0.0.1:5000/AcqgVf">http://127.0.0.1:5000/AcqgVf</a>	04.042023 15:00:34	7
3	<a href="https://sun9-69.userapi.com...">https://sun9-69.userapi.com...</a>	<a href="http://127.0.0.1:5000/HOoOiQ">http://127.0.0.1:5000/HOoOiQ</a>	04.042023 14:56:51	2
4	<a href="https://youtu.be/dQw4w9WgXcQ">https://youtu.be/dQw4w9WgXcQ</a>	<a href="http://127.0.0.1:5000/Gprebb">http://127.0.0.1:5000/Gprebb</a>	04.042023 14:55:17	2

## Задача 1

- 1 Создайте папку flask\_shortener.
- 2 Создайте и активируйте виртуальное окружение, установите **Flask**: `pip install flask`.
- 3 Создайте Flask приложение и добавьте маршруты:
  - ★ Главная страница '/' (функция `index`). Возвращает шаблон `index.html`.
  - ★ Страница с созданными ссылками '/urls' (функция `urls`). Возвращает шаблон `urls.html`.
  - ★ Эндпоинт '/<short>' (функция `url_redirect`). Перенаправляет пользователя по оригинальной ссылке.
- 4 Скачайте шаблоны `index.html` и `urls.html` и разместите в папке `templates`. На данном этапе изменять шаблоны не требуется.
- 5 Стили и скрипты подключаются в шаблонах, поэтому размещение статика не требуется.

## Задача 2

- 1 Измените структуру проекта, добавив наследование и переиспользование шаблонов: создайте базовый шаблон `base.html`, подключаемый шаблон с меню `navigation.html`, шаблоны с контентом, расширяющие базовый шаблон `index.html` и `urls.html`.
- 2 Сделайте рабочими ссылки меню, используя функцию `url_for()`.

## Дополнительно

Как сделать так, чтобы пункт меню становился активным на той странице, на которой находится посетитель?



## Задача 3

1 Установите flask-sqlalchemy:  
`pip install SQLAlchemy==1.4.7`  
`pip install Flask-SQLAlchemy==2.5.1`

2 Подключите SQLite базу данных `urls.db`, используя ORM.

3 Опишите модель `URLmodel` с полями:



`id` — первичный ключ



`original_url` (исходная ссылка) — строка (255 символов)



`short` (идентификатор короткой ссылки) — строка (255 символов)



`visits` (количество переходов) — целое число, по умолчанию значение 0



`created_at` (дата создания) — дата и время, по умолчанию текущая дата и время

4 Подключите тестовую базу данных `urls.db`.

5 Измените шаблон `urls.html` и выведите ссылки из базы данных.

## Задача 4

1 Установите Flask-WTF: `pip install flask-wtf`

2 Создайте форму `URLForm` с атрибутами:

✦ `original_url` — не пустое текстовое поле с текстом «Вставьте ссылку», с проверкой на то, что введена валидная ссылка. В случае ошибок валидации выводятся ошибки: «Ссылка не может быть пустой» или «Неверная ссылка».

✦ `submit` — кнопка отправки формы, текст «Получить короткую ссылку».

3 Напишите функцию `get_short()`, которая будет возвращать уникальную случайную строку (идентификатор для короткой ссылки) длиной шесть символов, состоящую из цифр, строчных или заглавных английских букв. Необходимо обязательно проверять, что такого идентификатора еще нет в базе данных, если есть, то получить новый.

4 Измените шаблон `index.html` для вывода полей и ошибок формы.

5 Измените функцию `index`, для обработки POST запроса с данными формы и создания новой записи в таблице базы данных. После создания записи пользователь перенаправляется на страницу с ссылками.

## Задача 5

- 1 Измените функцию `url_redirect`, чтобы при переходе по короткой ссылке, пользователь перенаправлялся по оригинальной ссылке.
- 2 При открытии короткой ссылки, у этой записи должен увеличиваться счетчик количества переходов и обновляться в базе данных.
- 3 Корректные значения количества переходов должны отображаться на странице с созданными ссылками.