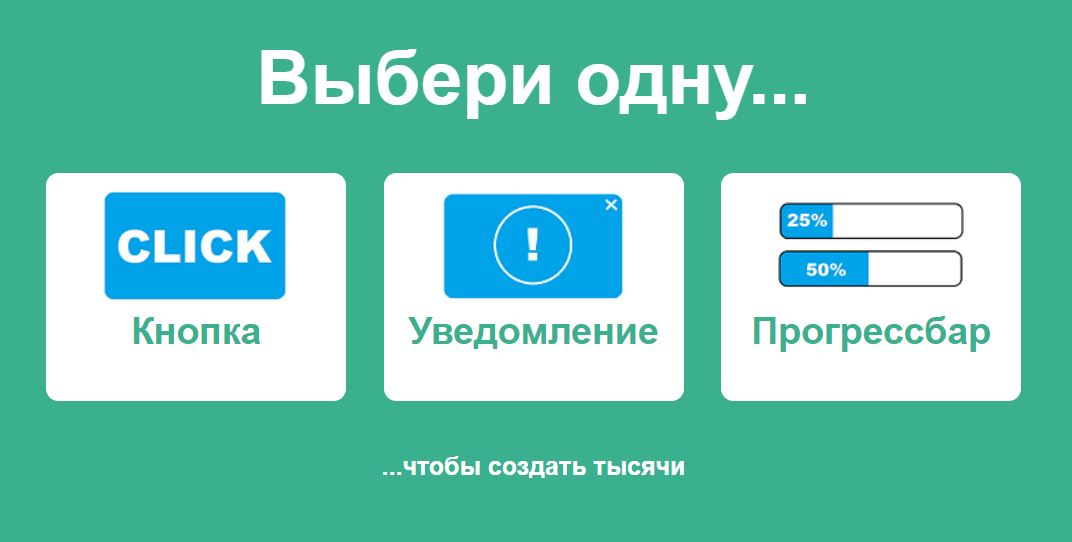
**Проект "Bootstrap-constructor"**

Разработчики: Никонова Наталья, Малоземова Анна

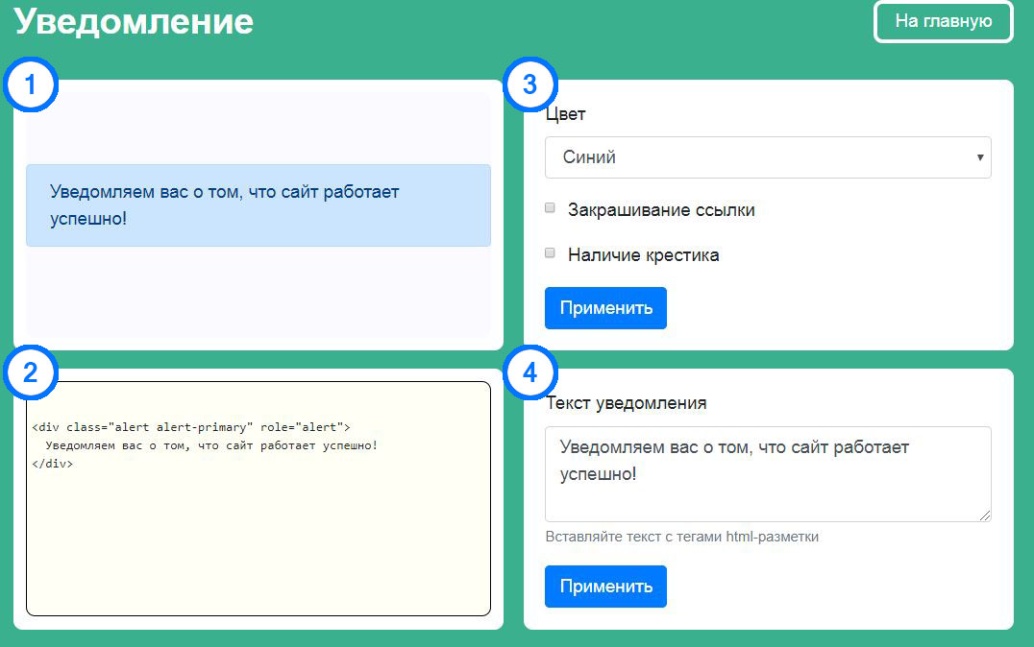
Ссылка на сайт: <https://bootstrap-constructor.herokuapp.com/>

Проект "**Bootstrap-constructor**" сделан с целью упрощения процесса создания bootstrap-объектов начинающими пользователями. Вместо самостоятельного изменения в html-коде параметров объекта, свойства и названия которых не всегда могут быть понятны новичку, он может воспользоваться интерактивным сайтом, выдающим html-код, описывающий объект с необходимыми пользователю параметрами.

На главной странице пользователь может выбрать один из трёх возможных объектов для создания: кнопка (button), уведомление (alert) или прогресс-бар (progress-bar). При дальнейшем развитии проекта планируется увеличить количество объектов, доступных для редактирования.

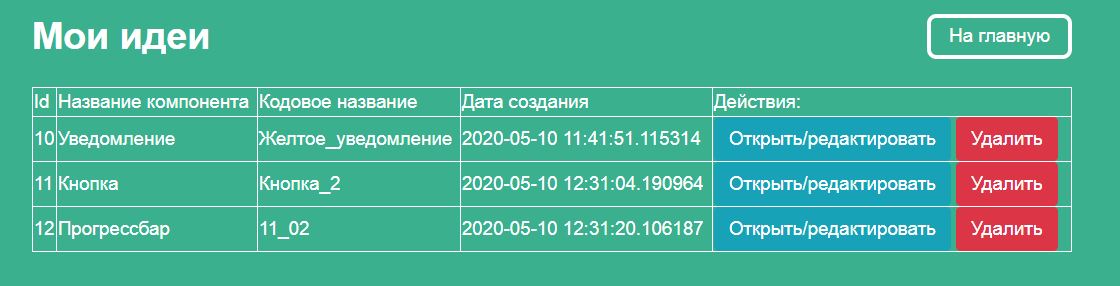


Кликнув по одному из объектов, пользователь попадает на страницу его редактирования. Она разделена на две части. Левая часть представляет внешний вид объекта с выбранными пользователем свойствами (1) и описывающий его html-код (2), в правой части показаны доступные для изменения свойства объекта (3, 4).



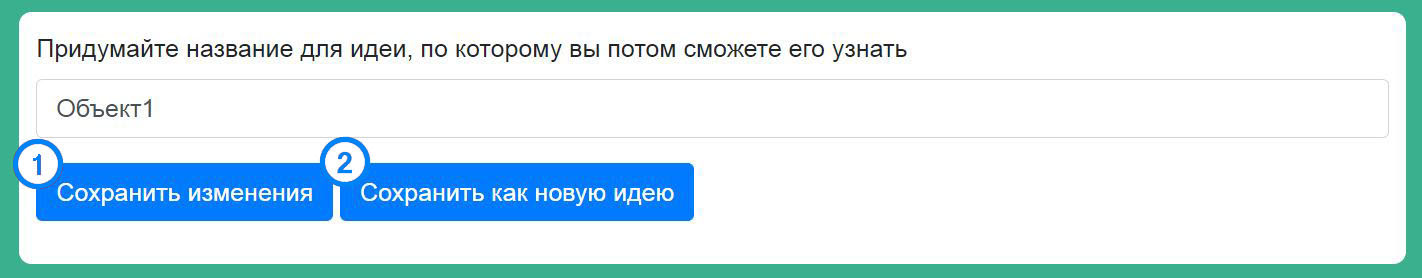
Выбрав в правой части страницы необходимые параметры, человек должен нажать кнопку "применить" и в левой части произойдут изменения объекта в соответствии с выбором свойств.

На сайте можно зарегистрироваться. Зарегистрированные пользователи могут сохранить созданные ими объекты. В личном кабинете пользователь может посмотреть сохранённые ранее "идеи", отредактировать их или удалить.



Внесённые изменения можно сохранить двумя способами:

1. Внести правки в старый объект.
2. Оставить старую идею неизменной, сохранив изменения как новую идею.



Для регистрации на сайте пользователю необходимо придумать логин, никнейм и пароль.

Для того, чтобы добавить полученный объект на свой сайт, пользователю нужно не только скопировать код, но и подключить сам Bootstrap.

**Использованные библиотеки:**

* certifi;
* chardet;
* chardet;
* Flask;
* Flask-Login;
* Flask-WTF;
* idna;
* itsdangerous;
* Jinja2;
* MarkupSafe;
* pip;
* requests;
* setuptools;
* SQLAlchemy;
* SQLAlchemy-serializer;
* urllib3;
* Werkzeug;
* WTForms.

**Структура программы**

В проекте были реализованы следующие классы:

* Object — для хранения информации об объектах: названия на английском и русском, списки изменяемых свойств объектов и их значений, служебные данные.
* Property — хранит названия свойств;
* Property\_values — хранит доступные для свойств значения в программном и понятном пользователю видах. Связан с соответствующим классом Property.
* User — хранит информацию о пользователе: логин, никнейм, хэшированный пароль, даты создания аккаунта.
* Idea — для сохранённых зарегистрированными пользователями идей. Связан с классами User и Object — содержит id сохранившего идею пользователя и id созданного объекта. Также хранит информацию о выбранных пользователем значениях свойств объекта, имя идеи и дату её сохранения.

На основе вышеперечисленных классов была создана база данных, состоящая из пяти таблиц.

Для получения данных из базы данных был написан код обработчиков API.

* get('http:// [bootstrap-constructor.herokuapp.com/](https://bootstrap-constructor.herokuapp.com/)api/users') — получение информации обо всех пользователях (за исключением хэшированного пароля);
* get('http://[bootstrap-constructor.herokuapp.com/](https://bootstrap-constructor.herokuapp.com/)api/users/id') — получение информации о пользователе с соответствующим id;
* delete('http://[bootstrap-constructor.herokuapp.com/](https://bootstrap-constructor.herokuapp.com/)api/users/id') — удаление из базы данных пользователя с соответствующим id;
* get('http://[bootstrap-constructor.herokuapp.com/](https://bootstrap-constructor.herokuapp.com/)api/ideas') — получение информации обо всех идеях;
* get('http://[bootstrap-constructor.herokuapp.com/](https://bootstrap-constructor.herokuapp.com/)api/ideas/id') — получение информации о идее с соответствующим id;
* delete('http://[bootstrap-constructor.herokuapp.com/](https://bootstrap-constructor.herokuapp.com/)api/ideas/id') — удаление идеи с соответствующим id;
* get('http://bootstrap-constructor.herokuapp.com/api/objects') — получение информации обо всех объектах;
* get('http://bootstrap-constructor.herokuapp.com/api/objects/id') — получение одного объекта по id;
* delete('http://bootstrap-constructor.herokuapp.com/api/objects/id') —удаление объекта по id;
* get('http://bootstrap-constructor.herokuapp.com/api/properties') — получение всех свойств;
* get('http://bootstrap-constructor.herokuapp.com/api/properties/id') — получение свойства по id.