|  |
| --- |
| *Автономная Некоммерческая Организация Дополнительного Профессионального Образования «Школа анализа данных» (АНО ДПО "ШАД") Подразделение "Лицей Академии Яндекса"(Яндекс.Лицей)* |
| **«Remarques»** |
| **Проект на языке программирования Python** |
|  |
|  |
| Авторы:  Чудинова М.А.  Пичугин А.Е.  Преподаватель:  Салтыков В. А. |
|  |

Нижневартовск, 2022

Оглавление

[Аннотация 3](#_Toc100939206)

[Первый запуск 4](#_Toc100939207)

[Реализация 5](#_Toc100939208)

[Заключение. 6](#_Toc100939209)

# Аннотация

Думая над темой для проекта мы вдохновились соц сетями и решили создать мини ленту публикаций от пользователей

Цель: Разработать сайт на flask

Задачи:

1. Изучить организацию подобных приложений.
2. Спроектировать пользовательский интерфейс.
3. Реализовать Взаимодействие с API и ORM моделями
4. Оформить проект.

Инструменты:

1. Python

Используемые модули:

* Sys
* Os

# Первый запуск

Репозиторий с проектом опубликован на GitHub по ссылке: <https://github.com/git21git/WEB_Project>

Главный файл для запуска проекта – «main.py»

Также программа работает по ссылке (Heroku):

# Реализация

Сайт состоит из нескольких страниц.

В работе использованы технологии:

* ORM модель (БД с публикациями)
* API Яндекс карт(для информации о странах и городах)
* Авторизация пользователя
* Компоненты css и bootstrap

**ORM модель**

У нас веб-приложение, в котором пользователи могут авторизоваться, просматривать, добавлять и удалять записи. В базе данных храним информацию о двух объектах:

* Пользователи (авторы)
* Записи (контент)

Публикация связана с пользователем, который ее написал.

**Работа с API**

В проекте было использовано API Яндекс карт для вывода изображений города и страны:

* Пользователь вводит тип карты и название города (страны).
* Направляется запрос к геокодеру на получение координат объекта.
* По координатам мы получаем изображение, сохраняем в файл и загружаем на страницу
* В случае ошибки на каком-либо этапе выполнения программы пользователь получит сообщение об ошибке и увидит карту Парижа.

**Авторизация пользователей**

На сайте возможна авторизация пользователей.

Как мы реализовывали регистрацию:

* Все классы для форм разместили в отдельном каталоге forms.
* Создали в каталоге forms файл user.py, а в нем — класс, описывающий форму RegisterForm, в которую добавили все поля, которые будут в форме регистрации.
* Спрашиваем пароль у пользователя несколько раз, чтобы убедиться, что он нигде не опечатался.
* После отправки формы на сервер проверяем, что пароли совпадают, а также что пользователя с таким адресом электронной почты пока нет в нашей базе данных.
* Если все хорошо, добавляем пользователя в базу данных и отправляем его на страницу авторизации

Реализация авторизации:

* Если форма логина прошла валидацию, мы находим пользователя с введенной почтой, проверяем, введен ли для него правильный пароль, если да, вызываем функцию login\_user модуля flask-login и передаем туда объект нашего пользователя, а также значение галочки «Запомнить меня».
* После чего перенаправляем пользователя на главную страницу нашего приложения.

**Компоненты css и bootstrap**

При оформлении дизайна сайта использовались компоненты bootstrap

# Заключение.

Цель проекта достигнута, сайт работает :)))

В качестве развития проекта можно предложить:

* - Улучшение дизайна сайта
* - Обмен личными сообщениями между пользователями
* - Добавление комментариев к записям
* - Поиск знакомых
* - Взаимосвязь аккаунтов

*но это уже другая история…*