МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Дополнительное образование «Основы промышленного программирования»

Python Flask проект на тему

«Web template configurator»

|  |  |  |
| --- | --- | --- |
| Выполнили: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Десятириков Ф.А. |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Бершова М.А. |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Гулимов М.А. |
| Методист: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Копытина Е.А. |

Воронеж 2021

# Содержание

[Содержание 2](#_Toc69149126)

[1 Актуальность темы 3](#_Toc69149127)

[2 Постановка задачи 3](#_Toc69149128)

[3 Авторы проекта 3](#_Toc69149129)

[3 Описание идеи 4](#_Toc69149130)

[4 Описание реализации 4](#_Toc69149131)

[5 Описание технологий 6](#_Toc69149132)

[6 Интерфейс программы «Web template configurator» 8](#_Toc69149133)

[7 Заключение 9](#_Toc69149134)

# Актуальность темы

В наше время всё большую популярность набирают сервисы, помогающие пользователям создавать сложные web-приложения без написания кода и каких-либо углубленных компьютерных знаний. Для таких приложений свойственна гибкость конфигурации и простота создания web-приложения. Большинство таких проектов конфигурируется из отдельного web-приложения. Приложение «Web template configurator» же позволяет собирать приложения офлайн, на устройстве пользователя. Кроме прочих, безусловным преимуществом такого автономного принципа сборки является гарантия безопасности контента.

# Постановка задачи

Требуется выполнить проект под названием «Web template configurator» на фреймворке Flask, который будет обладать следующими функциональными возможностями:

- Возможность создания (конфигурирования) готового web приложения под нужды пользователя путём изменения файлов конфигурации.

- Возможность регистрации и логина пользователей.

- Возможность пользователям коммуницировать друг с другом на сконфигурированной платформе.

- Возможность создавать модераторов.

- Возможность модерации созданной экосистемы.

# 3 Авторы проекта

Авторами-исполнителями проекта являются Десятириков Феликс Андреевич, Бершова Мария Александровна и Гулимов Михаил Алексеевич, ученики второго года обучения проекта Яндекс-Лицей.

# Описание идеи

Большинство приложений-конкурентов – такие как Tilda (<https://tilda.cc/ru/>) – не предлагают интегрированных чат-комнат. Приложение «Web template configurator» же реализует экосистему чата с модерацией сообщений и удалением пользователей. Модерацию может осуществлять только администратор экосистемы. Для осуществления модерации реализована консоль администратора (её можно открыть если нажать на имя пользователя в header) и файл management.py, эти блоки позволяют удалять сообщения и банить пользователей.

Главный экран представляет собой несколько блоков с различной информацией и картой. Футер – контактная информация. Хедер – набор кнопок для перехода на части страницы.

Регистрация и логин – формы для связи с бекэндом. На этих страницах реализована защита сокрытие пароля путём замены его на точки.

Чат представляет собой список сообщений со скролом и форму для отправки новых сообщений. При наведении курсора на сообщение показывается его id.

Сборка приложения производится запуском файла (configurator.py) с аргументами конфигурации. Аргументы можно увидеть с помощью команды --help.

# Описание реализации

Программный продукт «Web template configurator» был разработан с помощью следующих классов.

- class User() – класс базы данных – хранит информацию о пользователе и его данных, а именно поля: id (id), имя (name), email (email), админ ли пользователь (is\_admin), хешированный пароль (hashed\_password), дата создания аккаунта (created\_data), забанен ли пользователь (banned). Также класс реализует геттеры и сеттеры к этим полям.

- class Message() – класс базы данных – хранит информацию о сообщении, а именно поля: имя пользователя, который его отправил (user\_name), id (id), администратор ли отправитель (is\_from\_admin), текс сообщения (text), дата создания (created\_date). Также класс реализует геттеры и сеттеры к этим полям.

- class RegisterForm() и класс LoginForm() – классы форм – служат для общения фронтэнда и бэкэнда.

Остальной функционал заключается в файлах и их функциях:

- configurator.py – файл-сборщик – реализует конфигурирования конечного web-сервиса.

- management.py – файл модерации – нужен для управления пользователями и их сообщениями.

- db\_session – файл, реализующий взаимодействие с базой данных средствами сессий SQLAlchemy.

- sockeio\_helper.js – файл взаимодействия с библиотекой SocketIO – в нём реализована пересылка сообщений от клиента серверу (клиент-серверное взаимодействие).

- map\_script.js – файл отображения карты.

- password\_toogle.js – файл, скрывающий пароль на регистрационной странице.

- div\_scroll.js – файл, реализующий скроллинг в окне сообщений.

- scroll\_helper.js –файл, позволяющий производить плавный скролл.

- main – основной файл проекта. Он реализует несколько методов –

- parse\_all() – парсинг json файла в словарь.

- server\_start\_data\_read() – парсинг порта сервера

- main() – основной запускной файл проекта. Инициализирует проект и базу данных.

- load\_user(), logout(), login(), register() – методы, реализующие логин, регистрацию и логаут.

- administrate() – метод модерации системы. Реализует консоль администратора.

- chat() – метод, реализующий чат-комнату для пользователей.

- handleMassage() – хендлер получения данных из chat\_page.html, реализует получение и парсинг данных, передаваемых сокетным взаимодействием.

# Описание технологий

В разработанной системе «Web template configurator» реализованы рассмотренные в ходе изучения блока Flask технологии:

- Взаимодействие с базами данных средствами SQLAlchemy.

- Работа с операторами Jinja 2

- Работа с хешированием паролей.

- Работа с .css и .html

- Работа с файлами .json

- Работа с Argparse

- Работа с Flask, Flask\_loging, Flask\_ngrok, flask\_socketio

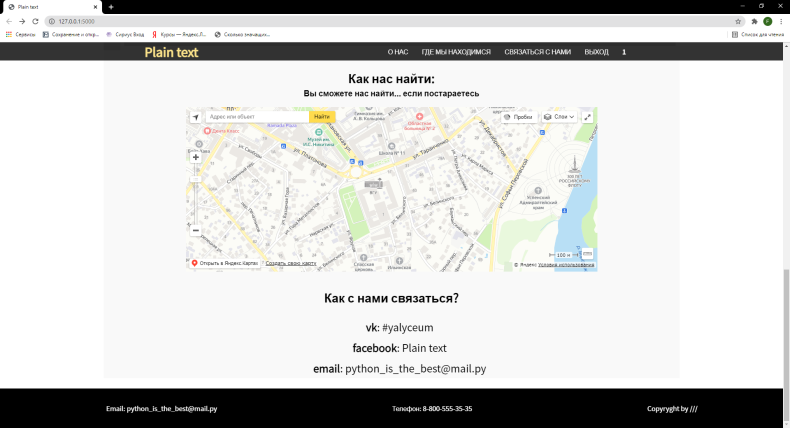
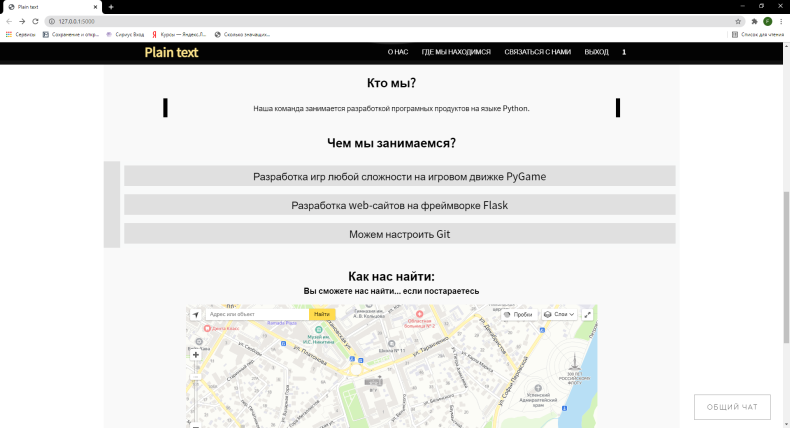
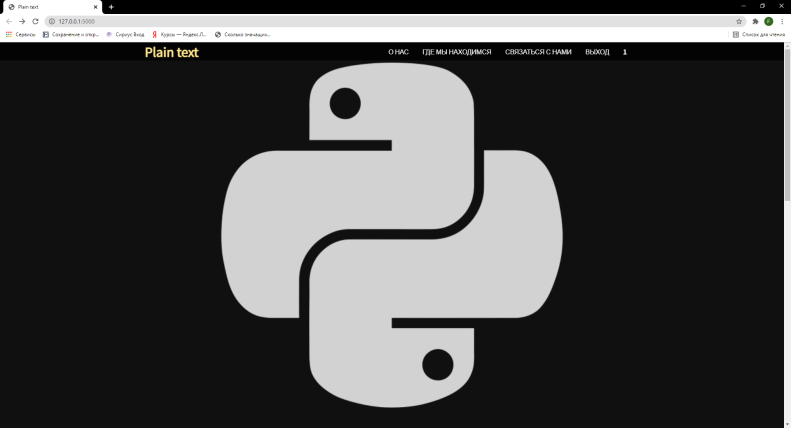
- Работа с JavaScript

- Работа с JQuery

- Работа с ymaps

Необходимые для запуска библиотеки представлены в файле requirements.txt

# Интерфейс программы «Web template configurator»

****

Рисуноки 1-3 - Главный экран

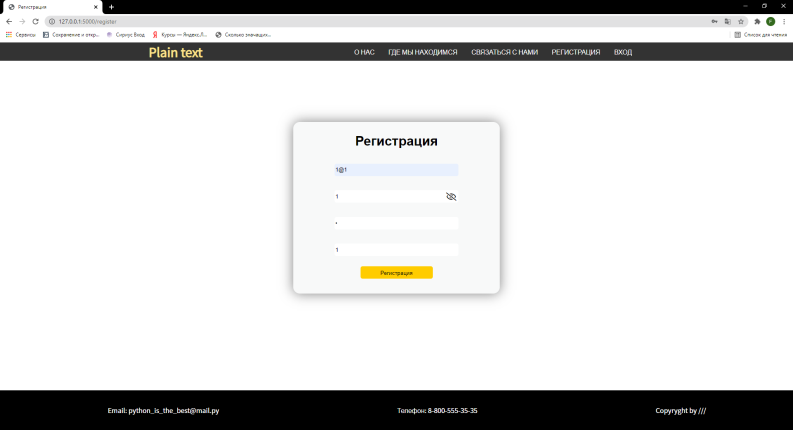


Рисунок 4 – Регистрация

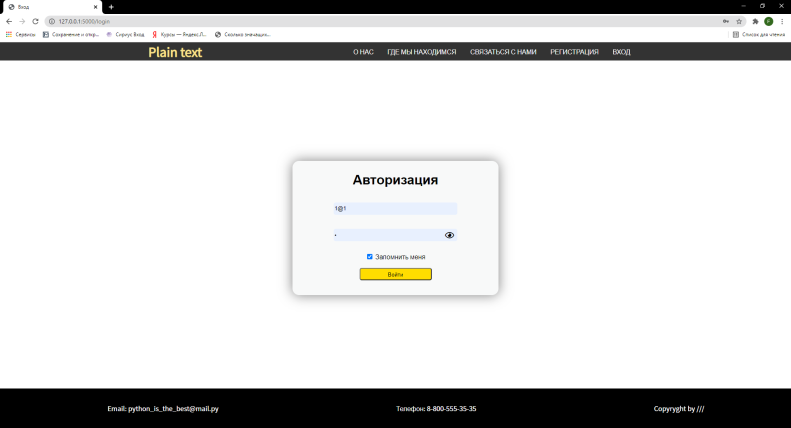
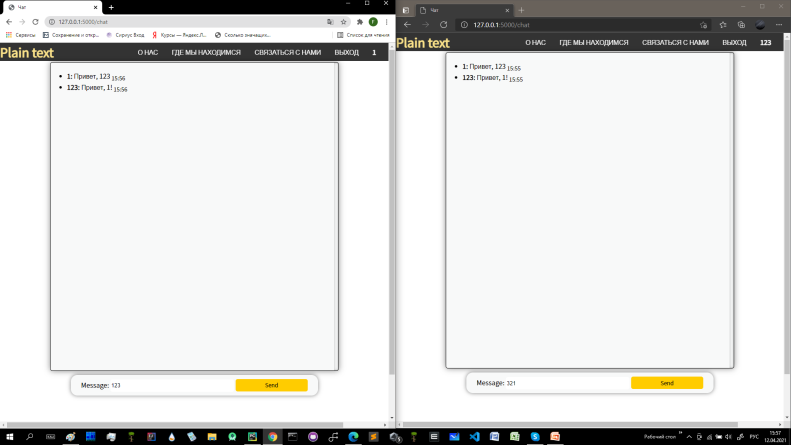


Рисунок 5 - Логин



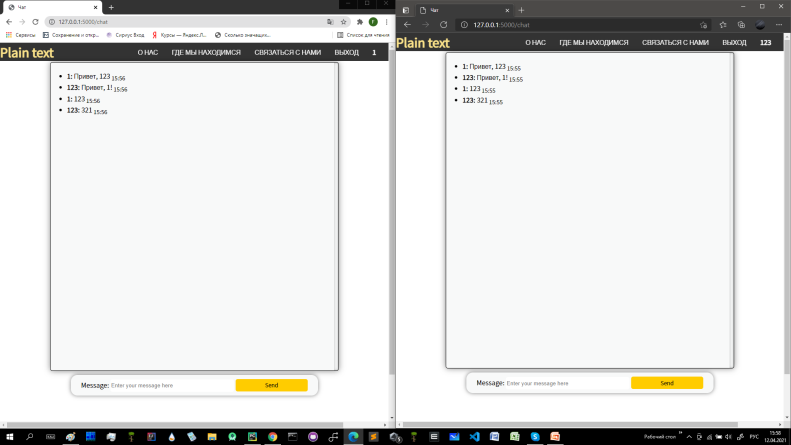


Рисунок 6 - Окно чата

# 7 Заключение

Средствами языка python, фреймворка flask и сторонних библиотек реализован коллективный проект, «Web template configurator». Програмный продукт предоставляет сервис создания web-приложений в офлайн режиме. Перспективными пользователями проекта являются физические и юридические лица, желающие создать деловое представительство в Интернете, и не обладающие при этом навыками продвинутого пользователя IT.