**School Bot**

**Авторы проекта**:

Семейкин Семён

Сергеев Данил

**Основные реализации**:

Для запуска бота используется файл main.py (school\_bot/bot/main.py).

Для запуска приложения flask используется файл index.py (school\_bot/index.py).

**Бот:**

В папке bot, хранится файл settings.py, в котором вы можете настроить сообщество, в котором будет работать бот, так же количество новостей, которые будут отправляться ботом по запросу пользователя.

Работа с ботом начинается с главного меню с тремя кнопками in-line клавиатуры: Расписание, Расписание звонков и Новости.

После выбора кнопок Расписание или Расписание звонков, бот предложит выбор дня недели, для которого вы хотите увидеть расписание, и кнопку возврата назад (в главное меню). Если вы выберете день недели, то следующим действием вам нужно будет ввести класс, для которого вы хотите увидеть расписание (если вы просматриваете расписание звонков, то этого действия не будет, расписание будет показано сразу). После выбора класса вы получите расписание для указанного класса, после чего будете возвращены в главное меню бота.

После выбора кнопки Новости, бот отправит вам последние N (указывается в настройках бота) новостей из базы данных.

**Дополнительные функции:**

* **send\_weekdays –** отправляет клавиатуру с выбором дней недели
* **send\_menu –** отправляет клавиатуру с главным меню бота

**Приложение Flask:**

Основная страница – страница с новостями. На ней отображаются все новости из базы данных. Если к новости прикреплен файл, то он так же отображается на этой странице около самого поля с новостью. Если файл является картинкой или видео файлом, то его можно просмотреть прямо на сайте, в остальных случаях по клику на название файла вы сможете скачать файл с сайта.

Слева располагается кнопка меню, по нажатию на нее появляется выпадающий список с навигацией по сайту. Возможны следующие переходы:

* Новости (основная страница сайта)
* Расписание уроков – после перехода по данной ссылке, вам будет предложено выбрать класс, из всех существующих в базе. После выбора класса вы увидите расписание на всю неделю для данного класса (или все существующие в базе дни недели), так же кнопку «Вернуться назад».
* Расписание звонков - после перехода по данной ссылке, вы увидите расписание звонков на всю неделю в виде таблиц для каждого дня, со столбцами: «Номер урока» и «Звонок».

**ЕСЛИ ПОЛЬЗОВАТЕЛЬ НЕ АВТОРИЗОВАН:**

* Войти в аккаунт – страница с формой входа в аккаунт системы. Так же есть возможность восстановления аккаунта в системе, путем отправки кода на email, прикрепленный к аккаунту.
* Зарегистрироваться – страница с формой регистрации, с полями: «Email», «Имя», «Фамилия», «Пароль» и «Повторите пароль».

**ЕСЛИ ПОЛЬЗОВАТЕЛЬ АВТОРИЗОВАН** (с правами администратора)**:**

* Добавление новостей – страница с формой добавления новости. На этой странице вы можете ввести заголовок и тело новости, так же прикрепить любой файл (картинка или видео имеют возможность быть просмотренными прям на странице новостей)
* Добавление расписания уроков – страница с формой добавления расписания уроков для отдельного класса, с полями: «Выбор класса» (число и буква), «День недели» и «Расписание», пример написания расписания можно посмотреть, нажав на соответствующую кнопку внизу формы.
* Добавление расписания уроков – страница с формой добавления расписания звонков. Вы можете выбрать дни недели, к которым будет относится данное расписание звонков, нажав на кнопку «выбрать дни недели» и проставив «галочки» в соответствующие поля, в теле расписания нужно указать звонки, пример написания расписания звонков можно посмотреть, нажав на соответствующую кнопку внизу формы.
* Пользователи – страница с пользователями системы, на которой вы можете изменять роль для пользователя или удалить пользователя, просмотреть всю информацию о пользователе (за исключением пароля). Никаких действий со своей учетной записью на этой странице выполнить нельзя.

**Использованные технологии**:

* datetime (Python библиотека)
* hashlib (Python библиотека)
* os (Python библиотека)
* uuid (Python библиотека)
* flask, flask-restful (Python библиотека)
* vk-api (Python библиотека)
* json (Python библиотека)
* request (Python библиотека)
* HTML
* CSS
* Работа с файлами любого формата
* Работа с базами данных sqlite

**Вывод:**

В ходе проделанной работы, мы смогли выполнить все поставленные цели, то есть создание бота и сайта для школы. В процессе выполнения работы мы сталкивались с различными трудностями, например, реализация плеера видео на сайте, которые мы успешно преодолели и сумели достичь поставленной цели.

Так же мы подразумеваем возможности доработки нашего проекта:

* Добавление функций бота и расширение его функционала
  + отображение уроков в реальном времени, например, «10:22 – сейчас перемена»
  + Публикация новостей с сайта на стену сообщества
  + возможность просмотра вложений к новости прямо из переписки с ботом
* Улучшение дизайна сайта.