Веб-сайт об истории и достопримечательностях города Пушкина

Проект Романенкова Арсения

Принцип работы веб-сайта

С первых секунд на сайте пользователя встречают поля для ввода почты и пароля. Если пользователь зарегистрирован на сайте и его данные есть в базе данных, то при вводе почты и пароля его отправляет на главную страницу, но если человек пользуется сайтом впервые, то для него есть специальная кнопка для регистрации, где он может придумать свой ник, пароль и зарегистрировать в базе данных свою почту.

После перехода на главную страницу пользователь может либо узнать больше об истории города Пушкина, либо о его Достопримечательностях. В правом верхнем углу также есть кнопка, которая при нажатии показывает данные профиля пользователя, под которыми он входил на сайт.

При переходе в раздел с историей города пользователь сможет узнать подробности происхождения Царского села, а также воспользоваться контекстными ссылками для перехода в определенный пункт, например “XX век”, в котором рассказывается о городе в советский период.

Описанное выше также можно увидеть при переходе в раздел достопримечательностей на главной странице. Отличается этот раздел картинками с достопримечательностями.

Структура проекта

1. Файлы с основным кодом .py:
   1. \_\_main\_\_.py – основной код, в котором находятся все декораторы и функции.
   2. \_\_all\_models.py – здесь хранятся модели для базы данных
   3. db\_session.py – отвечает за подключение к базе данных
   4. test.py – проверка работоспособности API
   5. user\_api.py – здесь хранится код для API
   6. users.py – здесь хранятся модели для базы данных
   7. user.py – здесь хранится модель входа в аккаунт для работы с базой данных
2. Шаблоны .html:
   1. base78.html – шаблон главной страницы
   2. base\_history.html – шаблон главной страницы с историей города
   3. buttons1.html – шаблон кнопок, наследованный от base78.html
   4. buttons\_history.html – шаблон контекстных ссылок, наследованный от base\_history.html
   5. buttons\_sightseeing.html – шаблон контекстных ссылок, наследованный от sightseeing.html
   6. login.html – шаблон, отвечающий за создание полей для входа в аккаунт
   7. profile.html – шаблон, отвечающий за создание профиля пользователя
   8. register.html – шаблон, создающий поля для регистрации аккаунта
   9. sightseeing.html – шаблон страницы с достопримечательностями
3. Декораторы и их функции:
   1. @app.route(‘/reg’, methods=[‘GET’, ‘POST’]) и def register() – создают страницу с регистрацией аккаунта, используя register.html
   2. @app.route(‘/base’, methods=[‘GET’, ‘POST’]) и def index() – создают главную страницу с помощью шаблона buttons1.html
   3. @app.route(‘/history’) и def history() – создают страницу с историей города с помощью buttons\_history.html
   4. @app.errorhandler(404) и def not\_found(error) – возвращает код ошибки 404
   5. @app.errorhandler(400) и def bad\_request(\_) – возвращает код ошибки 400
   6. @login\_manager.user\_loader и def load\_user(user\_id) – функция для получения данных пользователя
   7. @app.route(‘/’, methods=[‘GET’, ‘POST’]) и def login() – создают страницу для входа в аккаунт с помощью login.html
   8. @app.route(‘/sight’) – создают страницу с достопримечательностями с помощью buttons\_sightseeing.html
   9. def main() – создаёт базу данных и инициализирует работу сайта

База данных

В своём проекте я использовал базу данных blogs.db. В ней содержатся данные об аккаунтах пользователей, а именно их почта, зашифрованный пароль, никнейм пользователя, а также точное время создания самого аккаунта.

Вывод

В заключение, могу сказать что сайт еще дорабатывается, и возможно в скором времени будут добавлены комментарии, чтобы пользователи могли общаться друг с другом, а также множество других различных функций.