**Проект “Сайт для гостевых комнат ArtHouse Rooms”**

Выполнила Валерьянова А.

Идея

* Один из главных сервисов бронирования booking.com ушел из России.
* Гостевые комнаты имеют нестандартные номера, которые не поддерживаются обычными сервисами (каждый номер именованный, нет стандартного деления на простой и улучшенный), что вызывает неудобство и у клиентов, и у владельцев.
* У меня имелся опыт разработки бота, и хотелось попробовать что-то кардинально новое

Функциональная часть

**Языки программирования:**

1. Python – общее функционирование сайта

* project.py – главный файл
* email\_send.py – файл с функцией для отправки сообщений на email пользователя
* password.txt – файл с паролем от почты-отправителя (arthouserooms@gmail.com)

1. HTML – структура сайта

* base.html – файл-основа, подключает все стилевые файл и скрипты, содержит верхнее меню и footer.
* index.html – разметка главной страницы
* room.html – разметка для страниц всех номеров, содержит слайдер, в который передаются фото в зависимости от номера, и форму для поиска номера по дате.
* form.html – разметка для формы бронирования
* last.html – страница, появляющаяся после отправки формы бронирования, с благодарностью и контактным номером

1. CSS – дизайн сайта

* style.css – стили для base.html, переменные для используемых цветов
* indexstyle.css – стили для index.html
* room.css – стили для room.html, form.html, last.html

1. JavaScript – анимация (навигация (верхнее меню), запрет бронирования предыдущих дат и установка рамок бронирования в форме поиска номеров)

* main.js – открытие-закрытие верхнего меню
* map.js – взаимодействие с API Яндекс карт

**Используемые библиотеки:**

flask (Flask, render\_template, redirect, url\_for, flash)

flask\_sqlalchemy (SQLAlchemy) – взаимодействие с базой данных

datetime

pandas – для получения списка всех дат между двумя заданными

smtplib – для отправки email

email.mime.text (MIMEText) – для формирования сообщения для отправки по email

**Реализация программы:**

База данных:

Создана база данных, включающая 11 таблиц: 1 (profile) для сохранения всех данных о пользователе (из формы бронирования), остальные 10 – именованные таблицы номеров, записывают один параметр дату с ограничением на уникальность.

Страницы:

Создано 13 страниц: /main – главная, 10 именованных – для номеров, /book – форма бронирования, /success – страница успешной отправки формы

Именованные страницы:

Если тип запроса – GET, возвращают шаблон (render\_template) с названием номера на русском (для заголовка), на английском (для отображения фотографий именно этого номера в слайдере), расширением фотографии.

Если тип запроса – POST, проверяют корректность введенных дат (дата заезда должна быть раньше даты выезда) с помощью функции list\_date. Если все хорошо, функция возвращает список дат между двумя заданными, иначе строку с сообщением для пользователя, которая передается в flash, а затем снова возвращается шаблон. Далее в блоке try - except имитируется запись дат из списка с помощью db.session.flush(), если ошибки не возникает, нас перенаправляют на страницу формы бронирования, иначе даты – неуникальны, то есть уже заняты, db.session.rollback() и возвращается шаблон с flash('К сожалению, в эти даты номер уже занят').

Страница \book

Считается стоимость номера отеля на заданные даты по словарю со стоимостями номеров, а также по шаблону ниже:

if people == '2':  
 cost += (300 \* day)  
elif people == '3':  
 if name == 'origami':  
 cost += (1200 \* day)  
 else:  
 cost += (1000 \* day)

Если тип запроса – GET, возвращается шаблон страницы.

Если тип запроса – POST, в блоке try-except введенные пользователем данные записываются в таблицу profile. Возникли ошибки – db.session.rollback() и возвращается шаблон с flash. Ошибок нет – с помощью функции add\_date() записываем занятые даты в именованную таблицу. Формируем сообщение для отправки клиенту по email с помощью одного из html шаблонов и отправляем с помощью функции send\_email. После направляем на страницу /success с благодарностью за бронирование.

Дальнейшее развитие

* Подключение сайта к «единой шахматке бронирования», что позволило бы отслеживать бронирования с других сайтов.
* Создание страницы для написания отзывов, доступных только клиентам.
* Изменение цены в зависимости от сезона и ее высвечивание в форме поиска номера по дате.
* Добавление полной информации о каждом номере.
* Скрытие адресов страниц, на которых есть redirect.