

## **Разработка веб-сервиса для информационной системы «Удаленное управление температурой в помещении»**

*Радионов Сергей Владимирович*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

*Научный руководитель:*

*Баженов Руслан Иванович*

*Приамурский государственный университет имени Шолом-Алейхема*

*к.п.н., доцент, зав. кафедрой информационных систем, математики и правовой информатики*

### **Аннотация**

В данной работе описан процесс разработки веб-сервиса для информационной системы «Удаленное управление температурой в помещении». Представлен процесс разработки и реализации системы.

**Ключевые слова:** управление, Wi-fi, python, Django.

## **Development of a web service for the information system “Remote temperature control in a room”**

*Radionov Sergei Vladimirovich*

*Sholom-Aleichem Priamursky State University*

*Student*

*Scientific adviser:*

*Bazhenov Ruslan Ivanovich*

*Sholom-Aleichem Priamursky State University*

*Candidate of pedagogical sciences, associate professor, Head of the Department of Information Systems, Mathematics and Law Informatics*

### **Abstract**

This paper describes the process of developing a web service for the system "Remote control of room temperature". The process of developing and implementing the system is presented.

**Keyword:** management, Wi-fi, python, Django.

Достижение комфорта и энергоэффективности в современных системах отопления невозможно представить без качественных, надежных и понятных пользователю систем управления. Именно сегодня совершенствование этих систем является одним из ведущих направлений развития рынка теплотехнического оборудования. Особый интерес представляют системы

удаленного управления микроклиматом в помещении с помощью смартфона или компьютера посредством сети интернет.

В то же время, учитывая быстрый темп жизни, у современного человека далеко не всегда имеется возможность своевременно вносить необходимые изменения в работу приборов автоматического регулирования, например, изменять различные температурные режимы помещений. Ведь для этого необходимо находиться непосредственно на месте — в квартире или частном доме.

Именно поэтому в настоящее время в цивилизованных странах широкой популярностью пользуются устройства для удаленного управления системами отопления через сеть Интернет. Прикладные программы позволяют организовать простой доступ к системе через смартфон, ноутбук или планшет.

Актуальность задачи разработки веб-сервиса для информационной системы «Удаленное управление температурой в помещении» обусловлена необходимостью наличия сервера, через который будет осуществляться передача информации, а также её хранение в базе данных. Помимо этого, веб-сервис будет служить инструментом взаимодействия с пользователем для стационарных компьютерных систем. Поэтому и было решено разработать веб-сервис для системы «Удаленное управление температурой в помещении».

Для разработки веб-системы на языке программирования Python и фреймворке Django были рассмотрены различные литературные материалы. В.В. Моторин в своей работе рассмотрел сложности и нюансы программирования на языке python [1]. Сериализация и десериализация объектов python в yaml- формат была рассмотрена Н.А. Мясниковой, В.А. Шепиловым [2]. В своей работе Ю.А. Крапивко исследовал математические вычисления в языке программирования python [3]. В своем научном труде А.А. Агафонов и М.Р. Замалиев рассмотрели Создание динамического веб-сайта на языке программирования python с применением фреймворка django [4]. В своей работе Р.В. Ворожейкин описывает преимущества и недостатки использования объектно-реляционного отображения базы данных на примере фреймворка Django. Описывает понятие объектно-реляционного отображения и его достоинств, реализованном во фреймворке Django. Автор приводит список встроенных полей в ORM Django. Также приводит пример описания простой модели при помощи ORM Django и показывает как эта модель будет преобразована в SQL с синтаксисом PostgreSQL. Помимо этого, автор демонстрирует возможность выполнения SQL запросов средствами Django. Также демонстрирует возможность выполнения SQL-запросов в Django, несвязанных ни с одной моделью [5]. В статье А.Р. Исаева рассмотрены следующие вопросы: Django приложения, облегчающих жизнь при работе с данными; как организовывать код моделей в проекте; работу с чистыми SQL запросами; индексирования и поиск данных; унаследование в классах моделей; миграцию данных; дизайн базы данных и менеджеры паролей [6]. В работе К.М. Рябовой приведена архитектура фреймворка

Django, а также практические возможности программы [7]. Также не менее важны и иностранные работы [8-9].

Для взаимодействия пользователя с сервисом удаленного управления температурой в помещении и удаленным взаимодействием с устройством контроля температуры в помещении был разработан веб-сайт, на котором есть титульная страница (Рис.1), форма регистрации пользователя (Рис.2) и форма авторизации (Рис.3). Все формы проходят валидацию, если какое то поле заполнено неверно, то под полем красным цветом отображается ошибка, которую необходимо исправить, пример ошибок представлен на скриншоте формы регистрации (Рис.4).



Рис. 1. Титульная страница сайта

Рис. 2. Форма регистрации

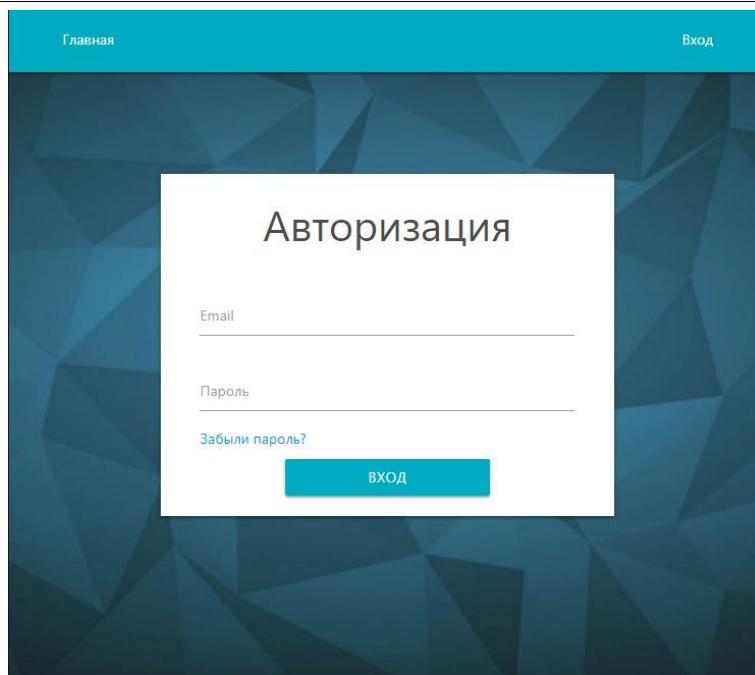


Рис. 3. Форма авторизации

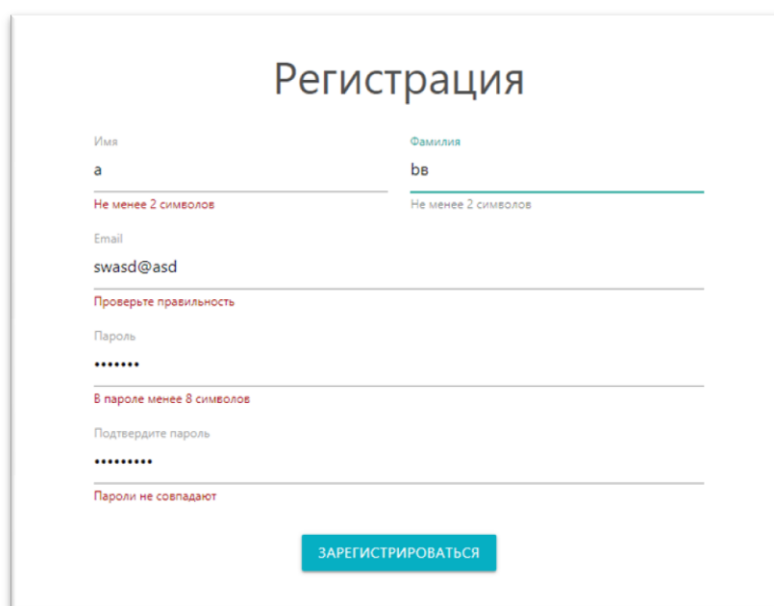


Рис. 4. Пример валидации формы регистрации

На сайте предусмотрена возможность восстановления пароля, если пользователь его забыл, для этого есть соответствующая форма (Рис.5). Восстановление пароля осуществляется с помощью отправки письма, содержащего ссылку восстановления в формате, на электронную почту пользователя (Рис.6). До отправки письма формируется `restore_token`, состоящий из 16 случайных букв латинского алфавита и сохраняется время, когда письмо было отправлено. Формат ссылки:

`<адрес сервиса>/account/restore/confirm/<id>/<restore_token>/`

где `id`– это `id`пользователя в базе данных, `token`– это `token`пользователя, состоящий из 16 случайных символов и хранящийся в БД. При переходе по этой ссылке на сервере происходит поиск пользователя по `id`, если такого

пользователя не существует отобразится ошибка 404. Если такой пользователь найден, `restore_token` в ссылке сопоставляется с `restore_token` пользователя, хранящимся в базе данных, также проверяется что Переход по ссылке произошел не более чем через 4 часа после отправки письма.

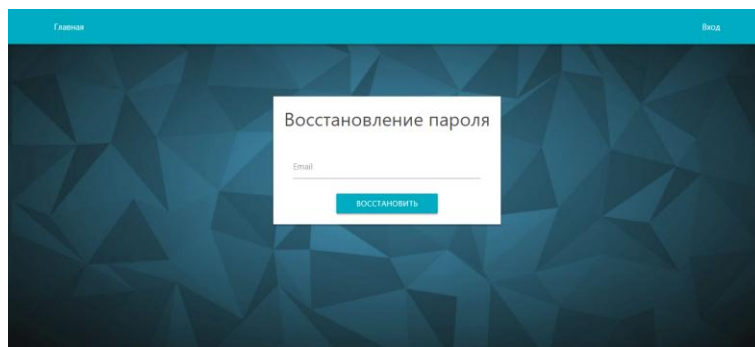


Рис. 5. Форма восстановления пароля

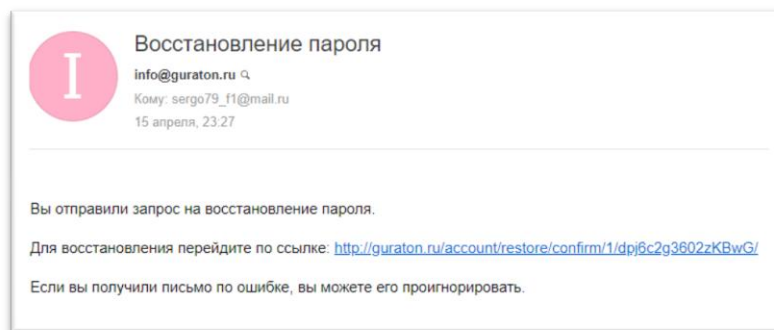


Рис. 6. Пример письма восстановления пароля

В веб-сервисе также есть личный кабинет (Рис.8), в котором отображаются данные с подключенных терморегуляторов, а также ключи для подключения терморегуляторов к сервису. Для того чтобы ключи, необходимые для подключения устройства к сервису можно было увидеть нужно подтвердить, что почта, на которую зарегистрирован аккаунт принадлежит пользователю. Для подтверждения отправляется письмо с ссылкой подтверждения на указанную почту (Рис. 7).

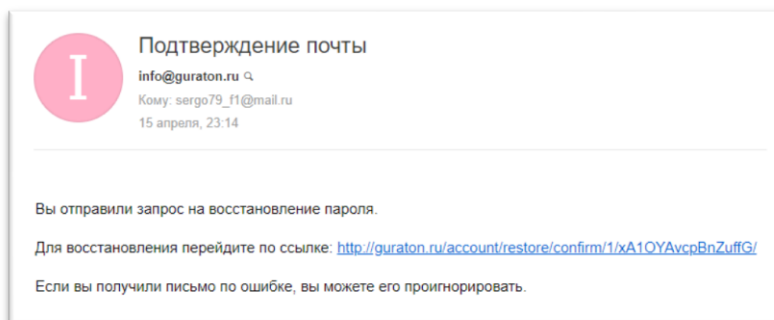


Рис. 7. Пример письма подтверждения почты

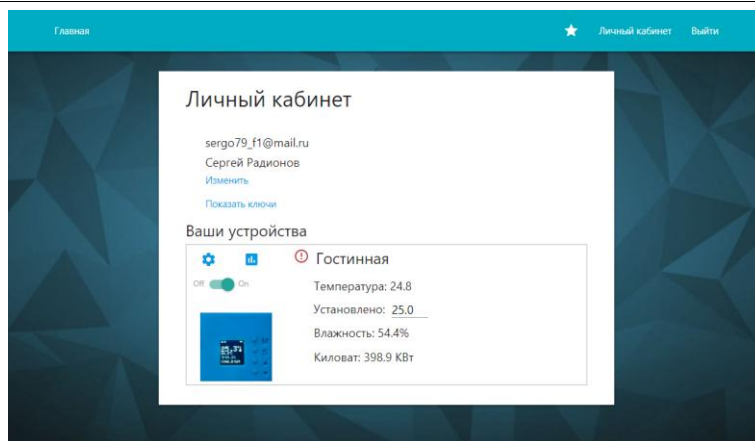


Рис. 8. Личный кабинет

В личном кабинете с помощью метода `load()` jQuery обновляются данные с датчиков с интервалом в 10 секунд без перезагрузки страницы. Если датчик в настоящее время не подключен к серверу отображается красный значок слева от названия датчика. Параметр «Установлено» является полем, которое можно редактировать, также можно включать следование заданной температуре и отключать переключателем в левом верхнем углу. После изменения этих параметров автоматически осуществляется запрос на сервер с новыми данными с помощью jQuery. Таким образом данные на сервере обновляются и сервер отправляет новые данные на терморегулятор.

Устройство соединяется с TCP сервером, который в свою очередь работает вместе с веб-сервером. Каждую минуту устройство отправляет данные на сервер в формате `(key1=value1&key2=value2...)`, где `key` – это название переменной, а `value` – её значение. Отправляемые данные содержат:

1. `String_id` – уникальный строковый идентификатор устройства. Если среди данных его нет или в БД нет устройства с таким идентификатором, то в таблице «`termocontrollers`» создается запись нового устройства, генерируется строковый идентификатор и отправляется устройству. Устройство при получении этого идентификатора перезаписывает свой на полученный.

2. `Api_key` – уникальный ключ, благодаря которому происходит привязка устройства к определенному пользователю.

3. `Model_name` – название модели устройства.

4. `Firmware_version` – название прошивки устройства.

5. `Temp` – текущая температура в помещении с устройством в градусах Цельсия, один знак после запятой.

6. `Target_temp` – целевая температура.

7. `Humidity` – текущая влажность в процентах, один знак после запятой.

8. `KWatts` – текущее значение счетчика потребления электроэнергии.

9. `Enable` – текущий статус устройства, включено поддержание целевой температуры или нет.

10. `ChangeEnable` – было ли изменено значение `enable` с помощью кнопок на устройстве, если да, то сервер перезапишет текущее значение в БД

на полученное с устройства, если нет, то при несовпадении значений сервер отправит текущее значение на устройство и оно перезапишет это значение.

11. `ChangeTargetTemp` – также как и `ChangeEnable`, только для целевой температуры.

12. `Hash` – содержит хеш сумму (SHA1) строки формата «<данные>; `Secret_key`». Это позволяет защитить данные от подмены и проверить источник отправляемых данных. `Secret_key` можно посмотреть только в личном кабинете пользователя и он вносится в настройки устройства с помощью приложения для смартфона. `Secret_key` не передается от устройства к серверу и наоборот. Сервер берет `Secret_key` из данных пользователя, который находится по `api_key`, далее происходит проверка хеш суммы, если хеш сумма не совпадает, то сервер считает, что полученные данные не валидны и ничего с ними не делает, а также отправляет устройству ошибку «`HashCheckError`». У устройства при получении такой ошибки отображается на дисплее иконка ошибки.

Сервер может отправить следующие данные устройству:

- `String_id`;
- `Target_temp`;
- `Enable` – значения `True` или `False`.

Эти данные отправляются по необходимости, также к ним прикрепляется значение `hash` для защиты.

Также можно посмотреть статистику температуры и влажности устройства (Рис.9) нажав на иконку статистики над картинкой устройства. Графики создаются с помощью JavaScript библиотеки `GoogleCharts` [16].

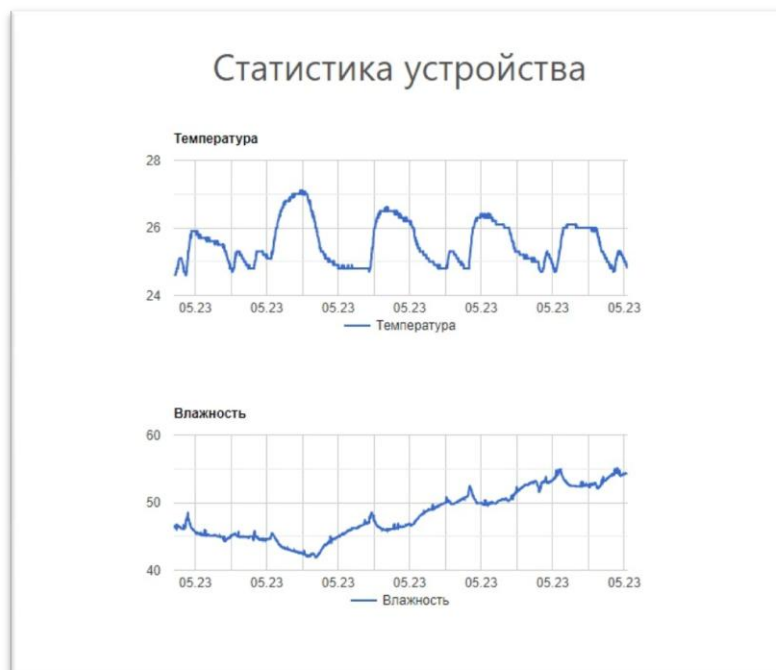


Рис. 9. Статистика устройства

Таким образом, был разработан веб-сервис для удаленного взаимодействия с устройством контроля температуры в помещении

**Библиографический список**

1. Моторин В.В. Программирование на языке python // Министерство образования и науки РФ ; Куйбышевский филиал ФГБОУ ВПО "Новосибирский государственный педагогический университет". Новосибирск, 2012.
2. Мясникова Н.А., Шепилов В.А. Сериализация и десериализация объектов python в yaml- формат // Объектные системы. 2014. № 8 (8). С. 52-57.
3. Крапивко Ю.А. Математические вычисления в языке программирования python // В сборнике: Наука и образование сборник статей X Региональной научной конференции. Ответственный редактор Е.Е. Адакин; Беловский институт (филиал) Кемеровского государственного университета; Администрация г. Белово Кемеровской области. 2013. С. 99-101.
4. Агафонов А.А., Замалиев М.Р. Создание динамического веб-сайта на языке программирования python с применением фреймворка django // В сборнике: Международная школа "Математическое моделирование фундаментальных объектов и явлений в системах компьютерной математики - KAZCAS-2016". Международная научно-практическая конференция "Информационные технологии в образовании и науке - ИТОН-2016" Труды школы и материалы конференции. Под общей редакцией Ю.Г. Игнатьева. 2016. С. 117-119.
5. Ворожейкин Р.В. Объектно-реляционное отображение базы данных на примере django // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2016. № 2 (12). С. 67-73.
6. Исаев А.Р. Как оптимально работать с базой данных и django моделями? // Электронный мультидисциплинарный научный журнал с порталом международных научно-практических конференций Интернетнаука. 2016. № 9. С. 127-135.
7. Рябова К.М. Фреймворк django: архитектура и возможности // В сборнике: СОВРЕМЕННЫЕ ТЕХНОЛОГИИ: АКТУАЛЬНЫЕ ВОПРОСЫ, ДОСТИЖЕНИЯ И ИННОВАЦИИ сборник статей XIII Международной научно-практической конференции: в 2 частях. 2018. С. 118-120.
8. Holovaty A., Kaplan-Moss J. The definitive guide to Django: Web development done right. Apress, 2009.
9. Sanner M. F. et al. Python: a programming language for software integration and development // J Mol Graph Model. – 1999. – Т. 17. – №. 1. – С. 57-61.
10. Радионов С.В. Разработка информационной системы «Удаленное управление температурой в помещении» // Постулат. 2019. № 6.