МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра радиотехники и связи

**Отчет по лабораторной работе № 6**

**“ Flask для визуализации данных InfluxDB"**

Выполнил: обучающийся гр. ИСТ – 23

Алексеев П.В.

Проверил: ст. преподаватель каф. РТиС Конкин Н.А.

Дата защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Йошкар-Ола

2025

**Цель:**

Получить базовые навыки работы с фреймфорком для решения задачи

разработки системы мониторинга и анализа данных с сенсорной сети использующей для хранения и обработки временных рядов.

Работа направлена на сбор хранение и визуализацию данных с сенсорной сети для обеспечения мониторинга и анализа

**Задачи:**

# Изучить и реализовать шаблон проекта.

# Вывести таблицу значений для графика временного хода.

# Выполнить получение данных метрик прогноза временного хода из InfluxDB а не из готового \*.csv файла.

**Теоретическая часть**

**Введение в Flask.**

Flask - это легковесный веб фреймворк на Python, который позволяет быстро и просто создавать веб-приложения. Он основан на WSGI (Web Server Gateway Interface) и Jinja2, что делает его гибким и расширяемым. Flask идеально подходит для небольших проектов и прототипов но также может быть использован для разработки более сложных приложений

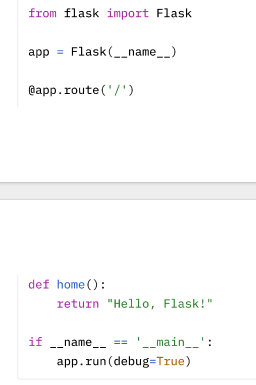
**Установка Flask**

Для начала работы с Flask, его необходимо установитью.Это можно сделать с помощью pip:

* pip install Flask

**Создание простого приложения**

Ниже представлен пример веб приложения на Flask:

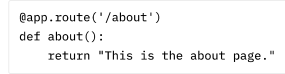


Описание кода:

1. Импортируем Flask: импортируем класс Flask из библиотеки Flask.
2. Создаем экземпляр приложения: app = Flask(\_\_name\_\_) создает экземпляр приложения.
3. Определяем маршрут: С помощью декоратора @app.route(‘/’) определяем маршрут для главной страницы. Функция home() возвращает текст “Hello, Flask!”.
4. Запускаем сервер: app.run(debug=True) запускает встроенный сервер Flask. Параметр debug=True позволяет автоматически перезагружать сервер при внесении изменений в код.

**Добавление дополнительных маршрутов.**

Можно добавить несколько маршрутов:



Теперь если вы перейдете по адресу /about , вы увидите текст “This is the about page.”

**Использование шаблонов**

Flask поддерживает шаблоны с использованием Jinja2. Для этого создадим папку *templates* и дабавим туда файл *index.html*



Теперь изменим код приложения для рендеринга этого шаблона:



**Запуск приложения.**

Чтобы запустить приложение, сохраните код в файл. например *app.py* , и выполните его:



После этого откройте в браузере адрес <http://127.0.0.1:5000/> , и вы увидите страницу , созданную с использованием шаблона.

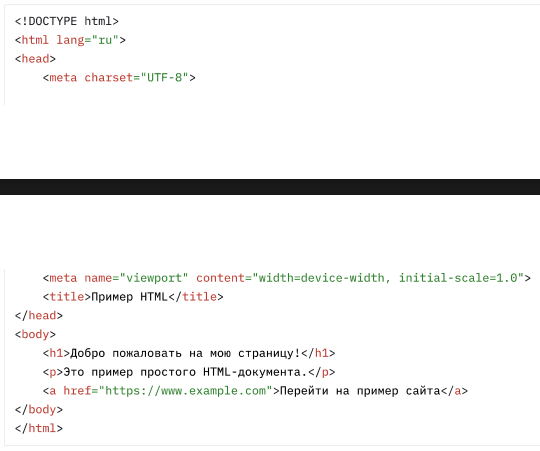
**Введение в HTML и CSS**

HTML (HyperText Markup Language) и CSS (Cascading Style Sheets) - это два основных языка, используемых для создания веб-страниц. HTML отвечает за структуру контента, а CSS- за его визуальное оформление.

HTML

HTML состоит из элементов, которые описываются с помощью тегов. Каждый тег может содержать текст, изображения, ссылки и другие элементы.

Пример простого HTML-документа:



В этом примере представлены основные элементы: заголовок ( <h1> ), параграф ( <p> ) и ссылку ( <a> ).

CSS

CSS используется для стилизации HTML-элементов. С помощью CSS можно изменить цвета,

шрифты, размеры и расположение элементов на странице.

Пример CSS-стилей:



Подключение CSS к HTML

CSS можно подключить к HTML-документу несколькими способами: встроенными стилями, внутренними стилями и внешними стилями.

Пример подключения внешнего CSS:



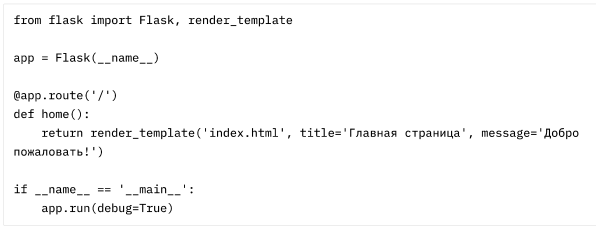
Теперь, если у вас есть файл styles.css с приведенными выше стилями, они будут применены к вашему HTML-документу.

HTML и CSS являются основой веб-разработки. С их помощью вы можете создавать простые и сложные веб-страницы, управляя как структурой, так и стилем контента.

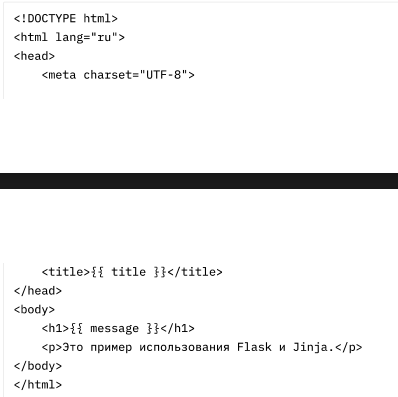
**Введение в использование Flask, Jinja и HTML**

Flask - это легковесный веб-фреймворк для Python, который позволяет быстро создавать веб-приложения. Jinja - это шаблонизатор, интегрированный в Flask, который позволяет динамически генерировать HTML-код, используя Python-переменные и логические конструкции.

1. Создание простого Flask - приложения



1. Создание HTML-шаблона (index.html):



Как это работает:

* •Flask создает веб-сервер и обрабатывает запросы к корневому URL ( / ).
* •Когда пользователь обращается к этому URL, вызывается функция home , которая использует render\_template для рендеринга HTML-шаблона index.html.
* Jinja заменяет переменные title и message на переданные значения,

создавая динамический HTML-контент.

**Практическая часть**

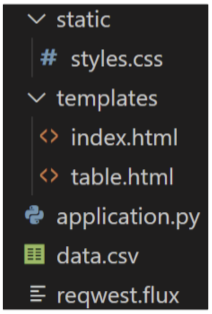
Структура проекта представлена на рисунке 1. style.css

таблица стилей для файлов разметки index.html и

table.html.

application.py - содержит исходный код формирования

web приложения средствами фреймворка Flask.

data.csv - временный файл с метками (для визуализации).

Рисунок 1. Структура учебного проекта Flask

request.flux - файл с запросом в базу данных InfluxDb.

На рисунке 2 представлен конечный вариант web страницы,

которая содержит графики реальных и прогнозных значений

сенсора, график и таблица полученных метрик прогноза.

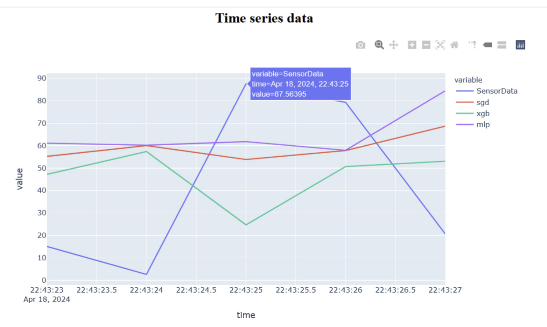
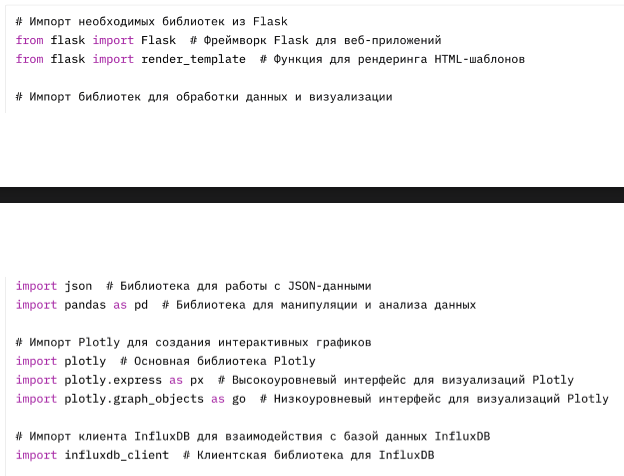


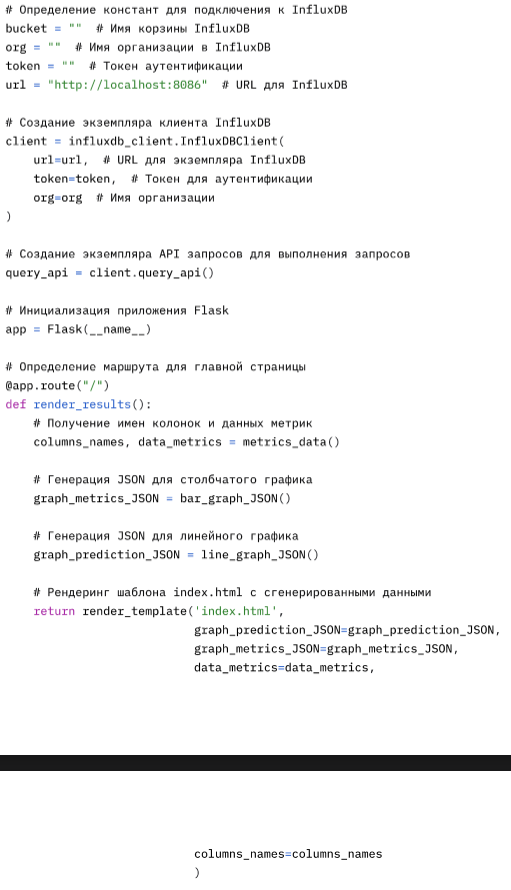


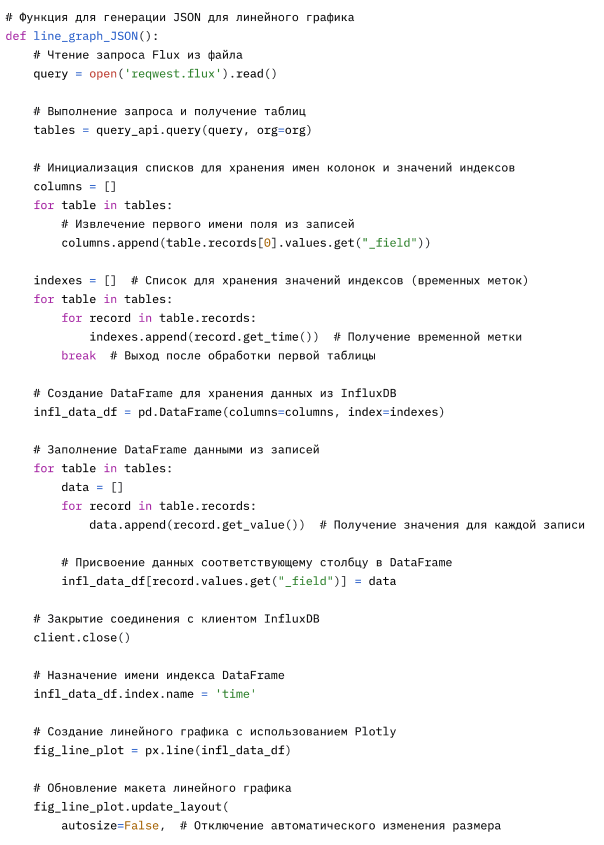
Рисунок 2. Внешний вид страницы с результатами прогноза и метриками по данным сенсорной сети Врехни график временных ходы рельного и прогнозных ходов данных сенсора Нижний график и таблица метрики прогноза

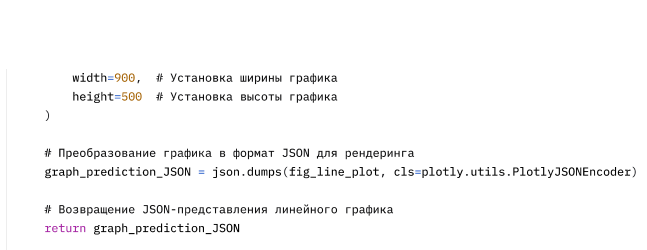
Далее реализуем структуру согласно рис.1.

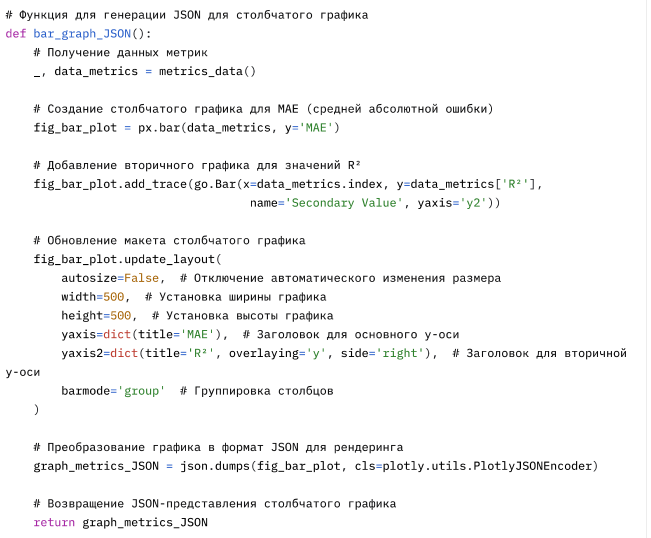
Структура файла application.py:

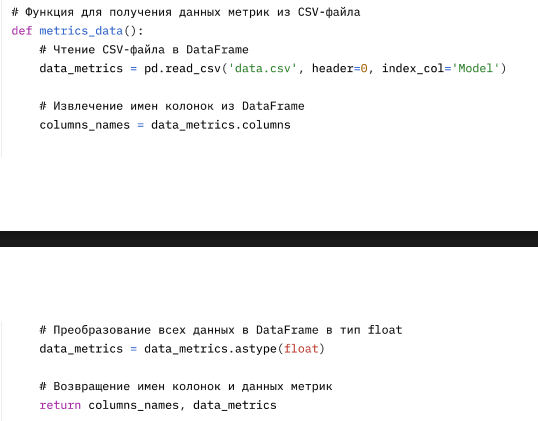










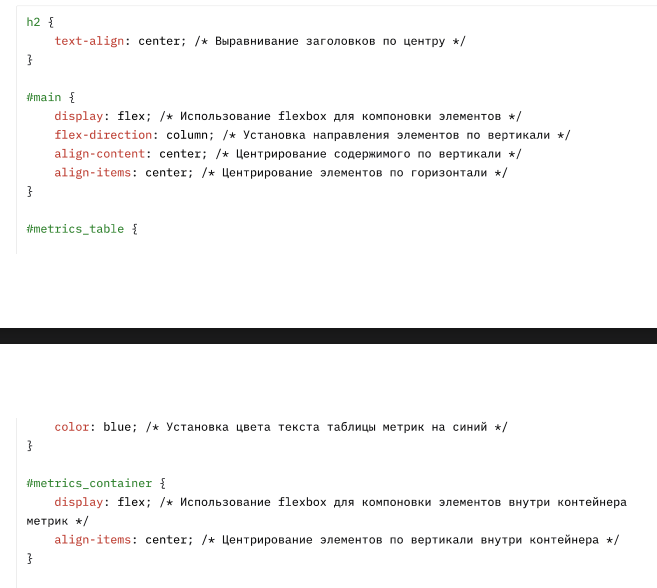




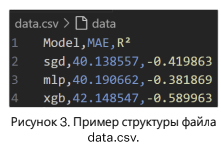


В файлах table.html и index.html в фигурных скобках используется функционал jinja2,который позволяет работать с передаваемыми через application.py данными, а именно интегрировать значения массивов, дата фреймов и переменных в структуру html файла.

Структура файлов styles.css:



Файл data.csv содержит метрики прогнозов используемых временных рядов, например (рис.3)



Таким образом будет выполнена задача №1.

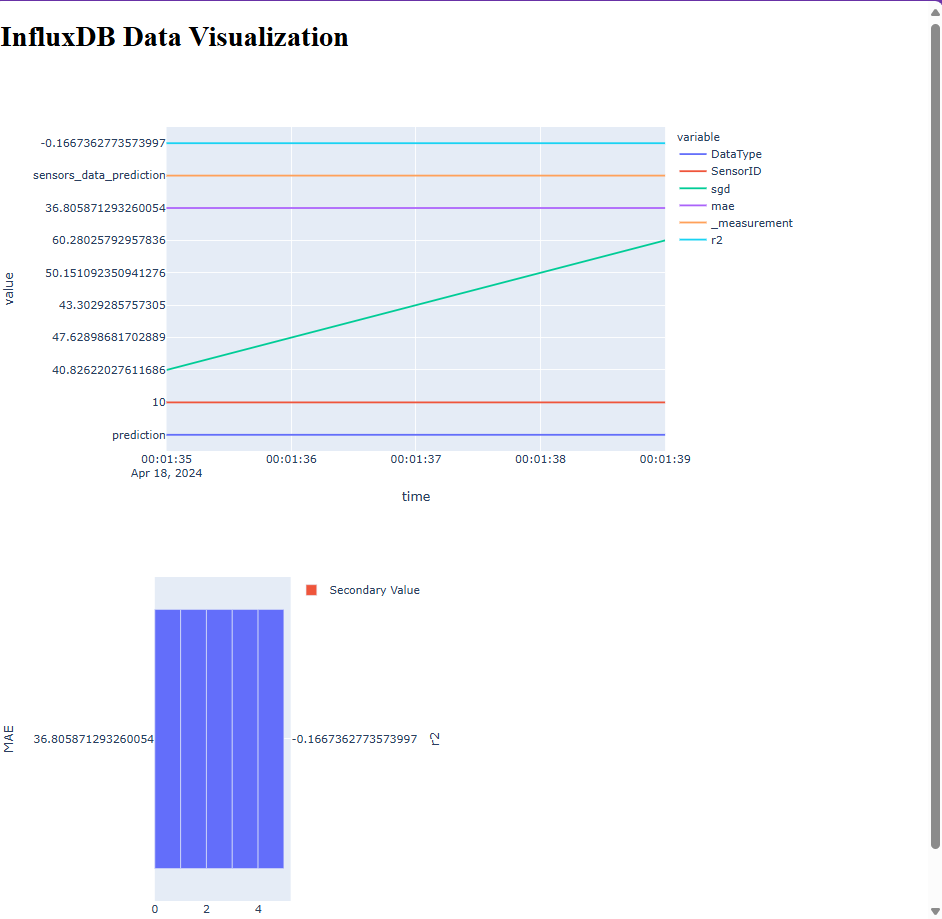
Для выполнения задачи №2 необходимо визуализировать

данные временных ходов (прогнозных и реального) в

формате html таблицы по аналогии с таблицей метрик.

Для выполнения задачи №3 необходимо получить из БД

InfluxDb результаты по метрикам МАЕ и К2 и передать их вместо данных файла data.csv.



**Вывод:**