Государственное бюджетное образовательное учреждение среднего профессионального образования Новосибирской области

Новосибирский химико- 630102 г. Новосибирск, ул. Садовая, 26,

технологический колледж. Тел./факс: (383) 266-00-44, тел.: (383) 266-00-54,

nhtk@mail.ru, http://nhtk-edu.ru

им. Д.И. Менделеева

**ПРАКТИЧЕСКАЯ РАБОТА**

по дисциплине «Обеспечение качества функционирования компьютерных систем»

Выполнила:

Здерева В.Э.

Новосибирск – 2024

nika.py

from flask import Flask, request  
import logging  
from logger import setup\_logger  
  
app = Flask(\_\_name\_\_)  
logger = setup\_logger()  
  
@app.route('/')  
def home():  
 logger.info('GET request to /')  
 return 'hi nhtk, i dont want go to vikapivat kartoshka!'  
  
@app.route('/data', methods=['POST'])  
def data():  
 logger.info('POST request to /data', extra={'body': request.json})  
 return 'Data received!'  
  
@app.errorhandler(Exception)  
def handle\_error(e):  
 logger.error('An error occurred: %s', e)  
 return 'Internal Server Error', 500  
  
  
@app.route('/favicon.ico')  
def favicon():  
 return '', 204 #(No Content)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(port=3000)

logger.py

import logging  
import socket  
  
#настройка UDP логгера  
class UdpHandler(logging.Handler):  
 def \_\_init\_\_(self, host, port):  
 super().\_\_init\_\_()  
 self.host = host  
 self.port = port  
 self.sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)  
  
 def emit(self, record):  
 msg = self.format(record)  
 self.sock.sendto(msg.encode(), (self.host, self.port))  
  
#создание логгера  
def setup\_logger():  
 logger = logging.getLogger('my\_logger')  
 logger.setLevel(logging.DEBUG)  
  
 #формат логов  
 formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')  
  
 #консольный вывод  
 console\_handler = logging.StreamHandler()  
 console\_handler.setFormatter(formatter)  
 logger.addHandler(console\_handler)  
  
 #файл логов  
 file\_handler = logging.FileHandler('nika.log')  
 file\_handler.setFormatter(formatter)  
 logger.addHandler(file\_handler)  
  
 #UDP логгер  
 udp\_handler = UdpHandler('166.168.1.100', 41234)  
 udp\_handler.setFormatter(formatter)  
 logger.addHandler(udp\_handler)  
  
 return logger

После запуска можно проверить логи в консоли и в файле nika.log.

Описание экосистемы:

Для выполнения задачи было выбрано простое веб-приложение на Flask —для разработки веб-приложений на Python.

Для логирования в приложении был выбран Python logging — встроенный модуль для логирования в Python.

Обоснование выбора:

* Стандартная библиотека: не требует дополнительных зависимостей, что упрощает установку и использование.
* Гибкость: поддерживает различные уровни логирования (DEBUG, INFO, WARNING, ERROR, CRITICAL).
* Поддержка форматов: можно настраивать формат сообщений.
* Расширяемость: легко интегрируется с другими системами и модулями.

В консоли:



И в nika.log

