МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина»

ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА

на тему

«Разработка web-приложение сервиса сбора отчетов от сотрудников о проделанных работах с оборудованием»

по дополнительной профессиональной программе –

программы профессиональной переподготовки

«Прикладное программирование на языке Python»

|  |  |
| --- | --- |
| Руководитель: Бутаков Руслан Александрович | подпись |
| Слушатель Брюханов Кирилл Вячеславович | подпись |

Екатеринбург

2024

**СОДЕРЖАНИЕ**

[**1. ПАСПОРТ ПРОЕКТА 4**](#_x98td6t7i93e)

[**2. ОПИСАНИЕ ПРОДУКТА ИЛИ УСЛУГИ 10**](#_p34hp4n7w2x9)

[**3. ОСНОВНЫЕ ЭТАПЫ/ИТЕРАЦИИ РЕАЛИЗАЦИИ ПРОЕКТА 11**](#_iisj561xvbn3)

[**4. ОРГАНИЗАЦИЯ РАБОТ В ПРОЕКТЕ 12**](#_l4x8nl4angcm)

[**5. РАЗДЕЛ ПО СПЕЦИФИКЕ ПРОГРАММЫ 13**](#_1blsro99oids)

[**6. ЗАКЛЮЧЕНИЕ 14**](#_jxnwbgip0b6y)

[**Библиографический список 15**](#_b94if0c2ruko)

# 1. ПАСПОРТ ПРОЕКТА

**Описание проекта**

Сервис сбора отчетов от сотрудников о проделанных работах с оборудованием. Сотрудникам предоставлена возможность загружать свои данные: fullname(имя и фамилия), machine(оборудование), work(проделанная работа) или просматривать другие отчёты сотрудников.

**Описание функциональности**

**1. Загрузка нового отчёта на сайт**

1.1. Загрузка нового отчёта:

Ввода определённой команды в Python Console - res =requests.post(url = "http://127.0.0.1:5000/add\_report", data = {'fullname': 'Nikitin Denis Alexandrovich', "machine": "Grinding machine", "work": "Setting processing speed"})

**2. Заполнение информации отчёта**

2.1. Ввод данных

Пример команды - res =requests.post(url = "http://127.0.0.1:5000/add\_report", data = {'fullname': 'Nikitin Denis Alexandrovich', "machine": "Grinding machine", "work": "Setting processing speed"})

В пункте 'fullname' – заполняется Фамилия и имя сотрудника

В пункте "machine" – заполняется название оборудования

В пункте "work" – заполняется проделанная работа

**3. Отправка данных на сервер**

3.1. После заполнения всех данных, при нажатии на кнопку клавиши «Enterь», все данные из формы отправляются на сервер <http://127.0.0.1:5000/> методом POST. Проверка загруженности данных можно увидеть во вкладке Run, если правильно загрузить данные, то высветится , иначе высветится ошибка.

**4. Просмотр загруженных отчётов**

4.1. Для просмотров отчёта нужно зайти на <http://127.0.0.1:5000> и вписать в конце одну из команд:

- /get\_report\_for\_id/<int:id> - поиск по id сотрудника.

Пример. [http://127.0.0.1:5000/get\_report\_for\_id/<1](http://127.0.0.1:5000/get_report_for_id/%3c1)>

Высветит на сайте первого по id сотрудника.

-get\_all\_reports

Пример. <http://127.0.0.1:5000/get_all_reports>

Высветит всю базу данных сотрудников загруженных ранее.

# 2. ОПИСАНИЕ ПРОДУКТА ИЛИ УСЛУГИ

Проект представляет собой веб-сервис для просмотра сбора отчетов от сотрудников о проделанных работах с оборудованием. Сотрудникам предоставлена возможность загружать свои данные или просматривать другие данные сотрудника, загруженные ранее.

Сотрудникам доступны следующие функции:

* загрузка свой данных;
* просмотр свои и других данных сотрудников;

Код проекта написан на языке Python. При проектировании архитектуры сервиса использовался классический принцип модульности — весь код, отвечающий за логику приложения, разделён на модули.

# 3. ОСНОВНЫЕ ЭТАПЫ/ИТЕРАЦИИ РЕАЛИЗАЦИИ ПРОЕКТА

1. Настройка рабочего окружения, знакомство с инструментами разработчика
2. Знакомство с основными возможностями Python — переменные, функции. Написание основных функций для проекта
3. Знакомство с основными возможностями Flask
4. Знакомство с основными возможностями SQLite
5. Организация кода — подготовка нужных модулей для проекта, создание точки входа
6. Понятие API и подключение сторонних Python -библиотек.
7. Оптимизация кода проекта, работа с данными и событиями.

# 4. ОРГАНИЗАЦИЯ РАБОТ В ПРОЕКТЕ

В свободное от учебы время, обычно это было вечером занимался работой по проекту, тимлид отвечал быстро, заняло около 3-4 вечера спокойно работ

# 5. РАЗДЕЛ ПО СПЕЦИФИКЕ ПРОГРАММЫ

Описание проекта:

1. Подключение библиотек и настройка Flask и SQLAlchemy.

2. Создание модели данных "Reportings" с определением структуры таблицы в базе данных.

3. Инициализация базы данных и создание таблицы Reportings.

4. Метод для добавления нового отчета в базу данных ("/add\_report").

5. Метод для получения отчета по id сотрудника ("/get\_report\_for\_id/<int:id>").

6. Метод для получения списка всех отчетов ("/get\_all\_reports").

7. Запуск приложения Flask при условии, что скрипт запускается напрямую (if name == "main").

Этот код позволяет создать простой веб-сервер с использованием Flask, который может принимать запросы на добавление отчетов, получение отчетов по id и получение списка всех отчетов. Модель данных представлена классом Reportings, который отображается на таблицу в базе данных.

Ссылка на репозитории *- https://github.com/KiryaTX/pythonProject*

# 6. ЗАКЛЮЧЕНИЕ

Я научился:

1. Импортировать необходимых модулей и библиотек: Flask, jsonify, request, Api, SQLAlchemy.

2. Создавать экземпляра приложения Flask.

3. Настроивать параметров базы данных SQLite.

4. Создавать объекта Api для работы с RESTful API.

5. Инициализировать объекта SQLAlchemy для работы с базой данных.

6. Определять модели Reportings для таблицы в базе данных. Свойства id, name, equipment, report.

7. Создавать метода init для инициализации объектов Reportings.

8. Создавать таблицы в базе данных с помощью db.create\_all().

9. Определять метода add\_report для добавления отчета в базу данных.

10. Определять метода get\_report\_for\_name для получения отчетов по имени сотрудника.

11. Определять метода get\_all\_reports для получения списка всех отчетов.

12. Запускать приложения Flask при условии, что имя модуля равно 'main'.

# Библиографический список

# <https://elearn.urfu.ru/course/view.php?id=6775> - Прикладное программирование на языке Python