Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Северо-Восточный федеральный университет имени М.К. Аммосова»
Институт математики и информатики

ДНЕВНИК
УЧЕБНОЙ ОЗНАКОМИТЕЛЬНОЙ ПРАКТИКИ

студента(-ки) второго курса группы ММ-ПМИ-23

Алексеевой Саргыланы Спиридоновны

Направление подготовки: 01.03.02 Прикладная математика и информатика
Направленность программы: Математическое моделирование и
вычислительная математика

Место прохождения практики: Дома самостоятельно.

Сроки практики: с 23 июня 2025 г. по 6 июля 2025 г.

Руководитель практики от кафедры ВТ: доцент, к.т.н. Акимов М.П.
должность, ФИО (полностью)

Якутск 2025

Содержание

СОДЕРЖАНИЕ ПРАКТИКИ

1. *Подготовительный этап*
   - Ознакомление с заданием и требованиями к отчету.
   - Изучение примеров дневников и отчетов.
   - Составление плана работы и выбора темы проекта.
2. *Основной этап*
   - Прохождение онлайн-курсов по SQL, Flask, REST API и тестирование в Postman.
   - Разработка веб-приложения «Калькулятор API» на Flask.
   - Написание тестов, подготовка документации и примеров запросов.
   - Размещение проекта на GitHub и деплой на хостинг.
3. *Заключительный этап*
   - Оформление отчета по практике.
   - Индивидуальные и групповые консультации с руководителем практики. Проверка подготовленных материалов.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ПРАКТИКИ

1) Повышение уровня практических навыков в области веб-разработки и программирования: овладение основными принципами работы с фреймворком Flask, создание REST API, разработка клиентской части на HTML/CSS/JavaScript, освоение основ взаимодействия фронтенда и бэкенда.
2) Приобретение опыта работы с инструментами тестирования и отладки веб-приложений, включая использование Postman для проверки API-запросов и автоматизации тестирования.
3) Развитие навыков работы с системами контроля версий Git и платформой GitHub: организация репозитория, ведение документации, оформление README.md с примерами запросов и инструкциями по запуску.
4) Получение опыта деплоя веб-приложений на облачные платформы, ознакомление с процессами развертывания и поддержки работающего прототипа.
5) Формирование умения самостоятельной организации рабочего процесса, планирования этапов разработки и соблюдения сроков выполнения проекта.
6) Закрепление навыков оформления технической и отчетной документации в соответствии с требованиями учебного заведения.
7) Содействие профессиональному самоопределению и ориентации в области IT-разработки через практическое выполнение полноценного проекта.

ЗАДАНИЕ УЧЕБНОЙ ОЗНАКОМИТЕЛЬНОЙ ПРАКТИКИ

- Согласовать план учебной практики и календарные сроки ее выполнения с руководителем практики.
- Вести дневник практики.
- По выбору обучающегося:
- Выполнение учебного IT-проекта «Калькулятор API»:
  - разработка веб-приложения на Flask с REST API;
  - создание веб-интерфейса для взаимодействия с калькулятором;
  - тестирование API с использованием Postman;
  - размещение проекта на GitHub с полной документацией;
  - деплой приложения на облачный хостинг.
- При прохождении практики подготовить отчет по практике, осветив в нем следующие основные вопросы:

  А) Описать выполнение заданий учебной практики;

  Б) Завершить отчет выводами и предложениями.

  В) При необходимости приложить к отчету по практике необходимые приложения

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Дата (период)	Содержание планируемой работы
23.06.2025-06.07.2025	1. Изучить онлайн-ресурсы и пройти онлайн-курс; 2. Сделать мини-проект; 3. Оформление дневника и отчета учебной практик;

РАБОЧИЙ ПЛАН (ГРАФИК) ПРОВЕДЕНИЯ ПРАКТИКИ

№ п/п	Этапы прохождения практики	Дата	Отметка о выполнении
1	Установочная лекция, инструктаж по общим вопросам, составление плана работы.	23.06.2025	выполнено
2	Ознакомление с заданием, выбор темы	24.06.2025	выполнено
3	Изучение Flask, REST API, основ SQL	26.06.2025	выполнено
4	Прохождение курса по Postman, получение сертификата	27.06.2025	выполнено
5	Разработка базовой структуры калькулятора	29.06.2025	выполнено
6	Реализация API: арифметика, факториал, корень	01.07.2025	выполнено
7	Добавление истории операций и веб-интерфейса	03.07.2025	выполнено
8	Написание тестов, проверка в Postman	04.07.2025	выполнено
9	Подготовка README, примеров запросов	05.07.2025	выполнено
10	Размещение на GitHub, настройка деплоя	06.07.2025	выполнено
11	Оформление отчета	06.07.2025	выполнено

В процессе прохождения практики студент не был допущен к сведениям, составляющим коммерческую тайну или нарушающим профессиональную этику.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Северо-Восточный федеральный университет имени М.К. Аммосова»

Институт математики и информатики

Отзыв-характеристика руководителя

о прохождении учебной ознакомительной практики

Студента Алексеевой Саргыланы Спиридоновны

Фамилия, имя, отчество

Группа: ММ-ПМИ-23 Курс: 2 Семестр: 4

Направление подготовки: 01.03.02 Прикладная математика и информатика

Направленность подготовки: Математическое моделирование и вычислительная математика

№ п/п	Критерии оценки	Оценка руководителя практики (по 5-балльной шкале)
1	Общая систематичность и ответственность работы в ходе практики	
2	Активность, самостоятельность, инициативность, творческая заинтересованность в ходе практики	
3	Эффективность организации взаимодействия с практикантом	
4	Полнота и качество выполнения работ, сформулированных в задании на практику	
5	Общий уровень сформированности компетенций	
6	Полнота и качество оформления отчетной документации	
7	Качество защиты отчета	
	ИТОГОВАЯ ОЦЕНКА*	

*Комментарии к оценкам:

_____

_____

_____

_____

Руководитель практики:_____/_____.

(подпись)

«____»_____20__ г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Северо-Восточный федеральный университет имени М.К. Аммосова»
Институт математики и информатики

ОТЧЕТ
ПО УЧЕБНОЙ ОЗНАКОМИТЕЛЬНОЙ ПРАКТИКЕ

Направление подготовки: 01.03.02 Прикладная математика и информатика
Направленность программы: Математическое моделирование и вычислительная математика

Студента группы ММ-ПМИ-23 Алексеевой Саргыланы Спиридоновны /
(группа, Ф.И.О / подпись)

Руководитель практики: доцент. к.т.н. Акимов М.П.
Оценка: _____/_____
(оценка, подпись)

Якутск 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Учебная ознакомительная практика является важным этапом в подготовке студентов IT-направления, позволяя применить теоретические знания в практической деятельности. В соответствии с заданием практики каждый студент должен самостоятельно выбрать тему IT-проекта, пройти соответствующие онлайн-курсы, разработать программный продукт и разместить его на GitHub.

Я выбрала тему «Разработка веб-приложения "Калькулятор API" на Flask. Данный проект предполагает создание серверной части на Python с использованием микрофреймворка Flask, реализацию REST API для выполнения математических операций и разработку простого веб-интерфейса для взаимодействия с пользователем.

Актуальность проекта обусловлена необходимостью получения практических навыков в области веб-разработки, создания API и работы с современными инструментами тестирования и развертывания приложений.

Целью практики являлась разработка полнофункционального веб-приложения калькулятора с возможностью выполнения основных арифметических операций, вычисления факториала и квадратного корня, ведения истории операций и предоставления доступа через REST API.

ОСНОВНАЯ ЧАСТЬ

1. Подготовительный этап

На подготовительном этапе была изучена документация по фреймворку Flask, основам работы с HTTP-запросами и ответами. Также был пройден онлайн-курс по тестированию API в Postman(Приложения), по итогам которого получен сертификат. Были определены основные функции приложения и составлен план разработки.

2. Процесс написания приложения

Разработка приложения велась поэтапно. На первом этапе была создана базовая структура Flask-приложения:

```python
import os
from flask import Flask, request, jsonify, render_template
import math

app = Flask(__name__)
history = []
MAX_LOG_SIZE = 50
```

Была объявлена функция для проверки целочисленного значения, необходимая для вычисления факториала:

```python
def check_integer(num):
    return isinstance(num, int) or (isinstance(num, float) and num.is_integer())
```

Далее был реализован главный маршрут, возвращающий HTML-страницу с интерфейсом калькулятора. Страница содержит кнопки для ввода цифр и операций, поле отображения вводимых данных и раздела истории операций. Основной функционал был вынесен в отдельный маршрут /calculate, который обрабатывает POST-запросы с данными в формате JSON. В зависимости от переданной операции выполняются соответствующие вычисления:

```python
@app.route("/calculate", methods=["POST"])
def calculate():
    data = request.json

    if not data or "op" not in data:
        return jsonify({"error": "Не указана операция"}), 400

    operation = data["op"]
    a = data.get("a")
    b = data.get("b")

    try:
        if operation == "+":
            result = a + b
        elif operation == "-":
            result = a - b
        elif operation == "*":
            result = a * b
        elif operation == "/":
            if b == 0:
                return jsonify({"error": "Деление на ноль"}), 400
            result = a / b
        elif operation == "^":
            result = a ** b
        elif operation == "sqrt":
            if a < 0:
                return jsonify({"error": "Корень из отрицательного числа"}), 400
            result = math.sqrt(a)
        elif operation == "fact":
            if a < 0 or not check_integer(a):
                return jsonify({"error": "Факториал только для целых >= 0"}), 400
            if a > 100:
                return jsonify({"error": "Слишком большое число"}), 400
            result = math.factorial(int(a))
        else:
            return jsonify({"error": "Неизвестная операция"}), 400

    except Exception as e:
        return jsonify({"error": f"Ошибка вычисления: {str(e)}"}), 400
```

После успешного вычисления результат операции сохраняется в истории:

```python
record = {"a": a, "b": b, "op": operation, "result": result}
history.append(record)

if len(history) > MAX_LOG_SIZE:
    history.pop(0)

return jsonify(record)
```

4. Тестирование API

После разработки основного функционала было проведено тестирование API с помощью Postman. Были проверены все типы операций, валидация входных данных и обработка ошибок. Создана коллекция тестовых запросов, которая включена в репозиторий проекта.

5. Размещение на GitHub и деплой

Проект был размещен на GitHub: https://github.com/Sargyl/Calculator-API.git.

Репозиторий содержит:

- Структурированный код;
- Есть README.md с описанием;
- Примеры API-запросов (curl/Postman);
- Тесты;
- Инструкции по локальному запуску и деплою;
- Демо проекта: ссылка на хостинг;
- Отчет по практике;

ЗАКЛЮЧЕНИЕ

В результате прохождения учебной ознакомительной практики было разработано веб-приложение «Калькулятор API» на Flask. В процессе работы были приобретены практические навыки back-end разработки, создания REST API, тестирования с использованием Postman и деплоя приложений.

Были успешно выполнены все поставленные задачи: реализован основной функционал калькулятора, создан веб-интерфейс, проведено тестирование, проект размещен на GitHub и развернут на хостинге.

Приложение может быть использовано как основа для более сложных проектов, связанных с веб-разработкой и предоставлением API-сервисов.

ПРИЛОЖЕНИЯ

Снимки экрана

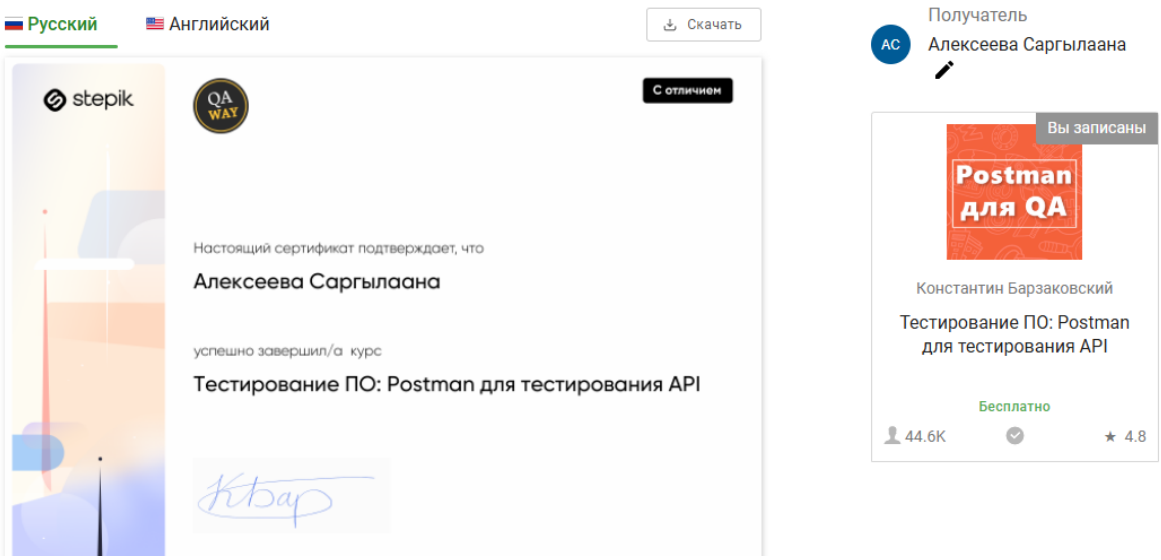

Рисунок 1. Прогресс онлайн-курса

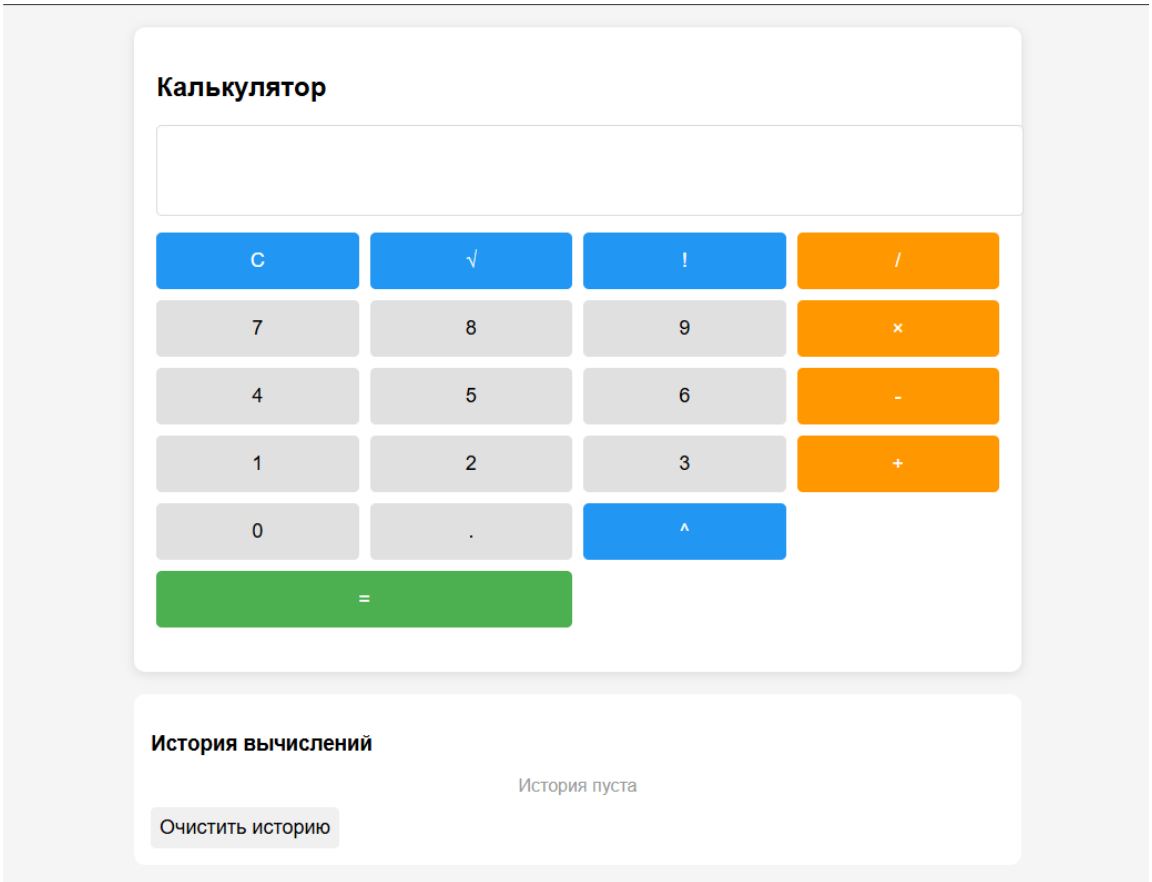

Рисунок 2.Главная страница калькулятора

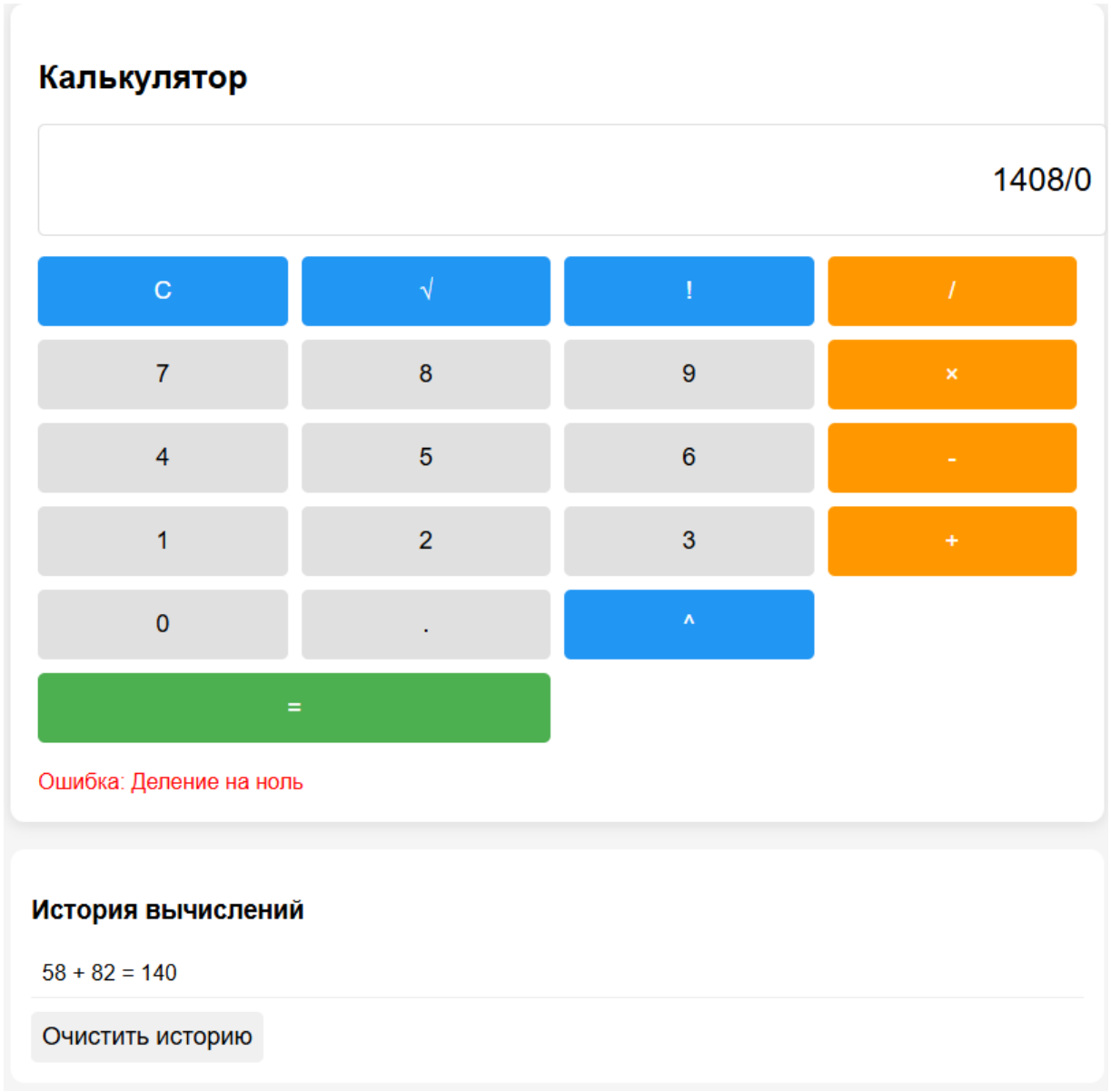

Рисунок 3.Пример ошибки и история вычислений

```python
import os
from flask import Flask, request, jsonify, render_template
import math

app = Flask(__name__)
history = []
MAX_LOG_SIZE = 50

def check_integer(num):
    return isinstance(num, int) or (isinstance(num, float) and num.is_integer())

@app.route("/")
def calculator():
    return render_template('calculator.html')

@app.route("/calculate", methods=["POST"])
def calculate():
    data = request.json

    if not data or "op" not in data:
        return jsonify({"error": "Не указана операция"}), 400

    operation = data["op"]
    a = data.get("a")
    b = data.get("b")

    try:
        if operation == "+":
            result = a + b
        elif operation == "-":
            result = a - b
        elif operation == "*":
            result = a * b
        elif operation == "/":
            if b == 0:
                return jsonify({"error": "Деление на ноль"}), 400
            result = a / b
        elif operation == "^":
            result = a ** b
        elif operation == "sqrt":
            if a < 0:
                return jsonify({"error": "Корень из отрицательного числа"}), 400
            result = math.sqrt(a)
        elif operation == "fact":
            if a < 0 or not check_integer(a):
                return jsonify({"error": "Факториал только для целых >= 0"}), 400
            if a > 100:
                return jsonify({"error": "Слишком большое число"}), 400
            result = math.factorial(int(a))
        else:
            return jsonify({"error": "Неизвестная операция"}), 400

    except Exception as e:
        return jsonify({"error": f"Ошибка вычисления: {str(e)}"}), 400

    record = {"a": a, "b": b, "op": operation, "result": result}
    history.append(record)

    if len(history) > MAX_LOG_SIZE:
        history.pop(0)

    return jsonify(record)
```

```python
@app.route("/log")
def get_log():
    return jsonify(history[::-1])

@app.route("/clear_log", methods=["POST"])
def clear_log():
    history.clear()
    return jsonify({"status": "История очищена"})

if __name__ == "__main__":
    port = int(os.environ.get("PORT", 10000))
    app.run(host='0.0.0.0', port=port, debug=False)
```

Рисунок 4.Код программы