**Министерство образования Иркутской области**

Государственное бюджетное профессиональное

образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

ПП.09.02.07-1.24.201.14

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

ПМ.02 Осуществление интеграции программных модулей

|  |  |  |
| --- | --- | --- |
| Руководитель от предприятия:  М.П. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (Е.Н. Дорохина) |
| Руководитель от техникума: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (С.Н. Касьяненко) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (А.А. Ларионова) |

Иркутск 2024

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc164474210)

[1 Создание проекта на GitHub 4](#_Toc164474211)

[2 Техническое задание на разработку базы данных 5](#_Toc164474212)

[3 Модуль приема данных 6](#_Toc164474213)

[4 Тестовые наборы данных для запросов GET и POST 9](#_Toc164474214)

[5 Настройка Linter 12](#_Toc164474215)

[ЗАКЛЮЧЕНИЕ 14](#_Toc164474216)

[Список используемых источников 15](#_Toc164474217)

## ВВЕДЕНИЕ

Одной из целей практики является создание Технического задания для разработки базы данных. Также разработка API, который будет использоваться для взаимодействия с базой данных. После этого проводится тестирование API с использованием специальных программ для проверки его работоспособности и корректности.

Наконец, результаты работы должны быть добавлены в репозиторий на платформе GitHub для общественного доступа. Таким образом, целью производственной практики является не только применение знаний, но и развитие навыков программирования, тестирования и работы с современными технологиями.

## Создание проекта на GitHub

Для создания проекта на GitHub необходимо создать новый или войти уже в существующий профиль. Также необходимо создать публичный проект (рисунок 1).

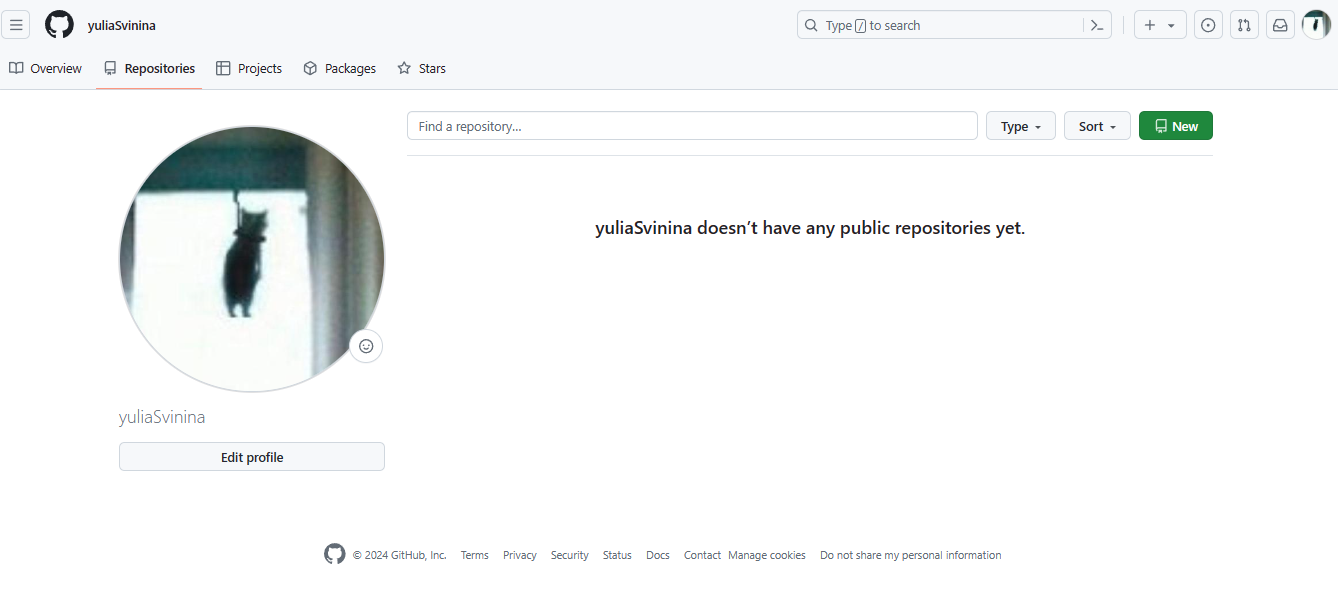


Рисунок 1 – Профиль GitHub

Ссылка на профиль GitHub: https://github.com/yuliaSvinina/PM02

## Техническое задание на разработку базы данных

Разработка технического задания была реализована в соответствии с ГОСТ 34.602-89 на тему «Каталог компьютерной техники».

ГОСТ 34.602-89 - это ГОСТ СССР, утвержденный в 1989 году и называется "Автоматизированные системы. Основные понятия". Он устанавливает основные термины и определения, используемые в области автоматизированных систем, включая термины, относящиеся к базам данных.

Некоторые из ключевых терминов и определений, которые могут быть включены в ТЗ по ГОСТ 34.602-89 для базы данных, включают следующее:

– База данных

– Таблица базы данных

– Поле базы данных

– Запись базы данных

– Ключ базы данных

– Связь таблиц базы данных

– Индекс базы данных

– Транзакция базы данных

– Уникальность данных

– Целостность данных

Таким образом, при разработке ТЗ для базы данных согласно ГОСТ 34.602-89, следует учитывать данные понятия и определения, чтобы обеспечить корректное понимание требований к базе данных и ее функциональности.

## Модуль приема данных

Для модуля приема данных был использован FastApi.

FastAPI – это современный веб-фреймворк для создания API на Python, который позволяет быстро и легко разрабатывать высокопроизводительные веб-приложения.

FastAPI обладает множеством преимуществ, включая:

– Автоматическую генерацию документации API на основе стандарта OpenAPI.

– Поддержку асинхронного программирования, что позволяет создавать высокопроизводительные приложения.

– Декларативный синтаксис, который позволяет определять маршруты и модели данных с минимумом кода.

– Автоматическую валидацию данных запросов и ответов.

– Поддержку множества асинхронных ORM (Object-Relational Mapping) и баз данных, таких как SQLAlchemy, Tortoise ORM, и др.

– Широкий набор инструментов для тестирования и отладки приложений.

FastAPI стал популярным выбором для разработки веб-приложений на Python благодаря своей производительности, удобству использования и интеграции с другими библиотеками и инструментами.

Для работы его необходимо установить, сделать это можно с помощью команды – pip install fastapi uvicorn. Результат представлен на рисунке 2.

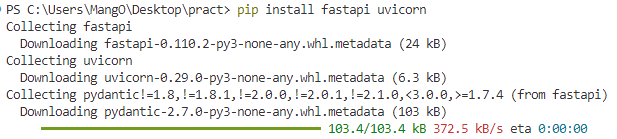


Рисунок 2 – Установка FastApi

Также необходимо установить модуль psycopg2 для работы с PostgreSQL (рисунок 3).

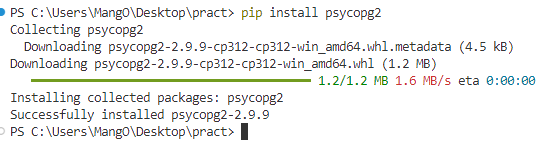


Рисунок 3 – Установка модуля для работы с PostgreSQL

Для работы с данными нужно использовать БД. Это реализуется благодаря БД. Были рассмотрены следующие варианты реализации СУБД:

1. SQLite;
2. MySQL;
3. PostgreSQL.

В качестве БД для будущего продукта была выбрана PostgreSQL, так как она масштабируема и предоставляет весть необходимый функционал для разработки продукта.

Это бесплатная открытая платформа управления базами данных, которая предоставляет удобный интерфейс для работы с различными типами баз данных, такими как PostgreSQL, MySQL, SQLite и другие. pgAdmin обеспечивает возможность создания, удаления и изменения баз данных, таблиц, индексов, хранимых процедур, а также выполнения SQL-запросов и мониторинга производительности баз данных.

Создаем базу данных и таблицу users для приема данных (рисунок 4).

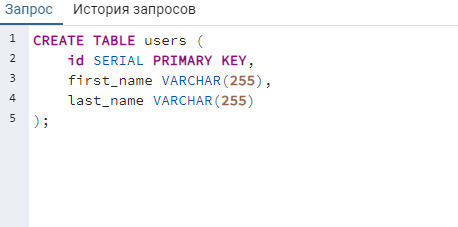


Рисунок 4 – Создание таблицы User

Далее представлен код для приема и обработки данных проверки на корректность и последующей записи в базу данных на основе PostgreSQL (рисунок 5).

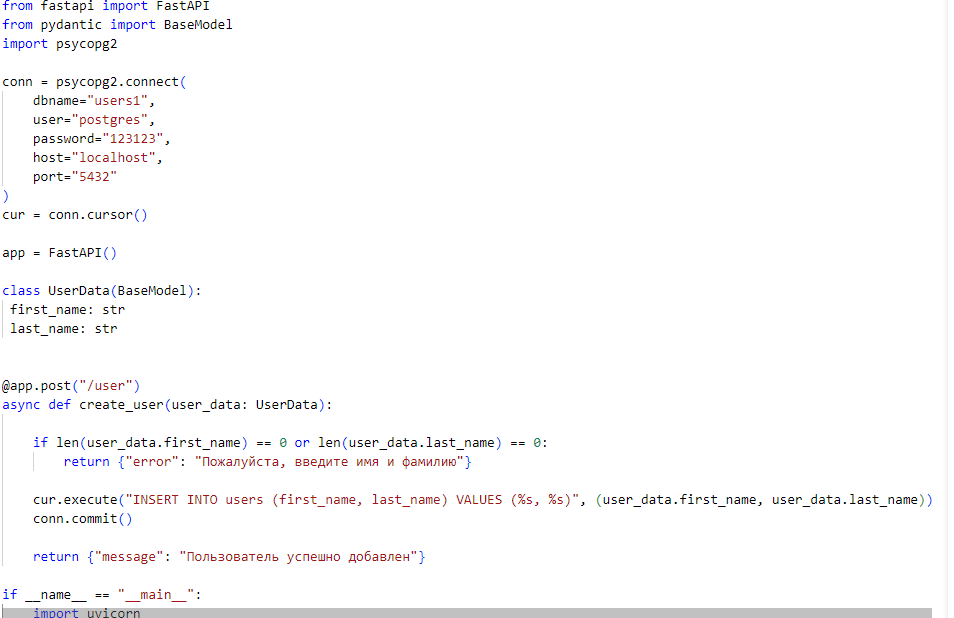


Рисунок 5 – Код для приема и обработки данных

Для реализации приема данных необходимо выполнить подключение к базе данных, далее прописать форму данных для POST–запроса.

Следующим этапом проверяется корректность введенных данных с последующим добавлением их в бд.

Для запуска сервера используется pyton pi.py (рисунок 6).

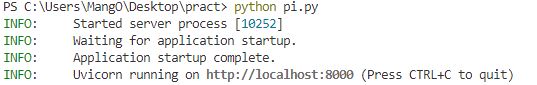


Рисунок 6 – Успешный запуск сервера

## Тестовые наборы данных для запросов GET и POST

Методы GET и POST являются основными методами HTTP запросов, которые используются для отправки данных на сервер.

1. Метод GET:

– Метод GET используется для запроса данных с сервера. Он передает параметры в URL запроса.

– Он является идемпотентным, что означает, что при нескольких одинаковых запросах результаты должны оставаться неизменными.

– Данные, передаваемые через метод GET, ограничены длиной URL и могут быть просмотрены в адресной строке браузера.

– Пример использования метода GET: получение списка пользователей, фильтрация данных и т.д.

2. Метод POST:

– Метод POST используется для отправки данных на сервер для создания или обновления ресурсов.

– Данные, передаваемые через метод POST, могут быть скрыты от просмотра (например, при отправке формы).

– Он не является идемпотентным, что означает, что при нескольких одинаковых запросах могут быть получены разные результаты.

– Пример использования метода POST: отправка данных формы, создание нового пользователя и т.д.

При разработке веб-приложений, сервер должен правильно обрабатывать как метод GET, так и метод POST, чтобы эффективно обрабатывать запросы от клиентов и взаимодействовать с базой данных или другими ресурсами.

Для реализации набора тестовых данных использовался Postman.

Postman – это инструмент для тестирования и разработки API. Он позволяет создавать, отправлять и тестировать HTTP запросы к различным эндпоинтам API, а также автоматизировать процессы тестирования API. Postman также предоставляет возможность сохранять запросы, создавать коллекции запросов и обмениваться ими с другими пользователями.

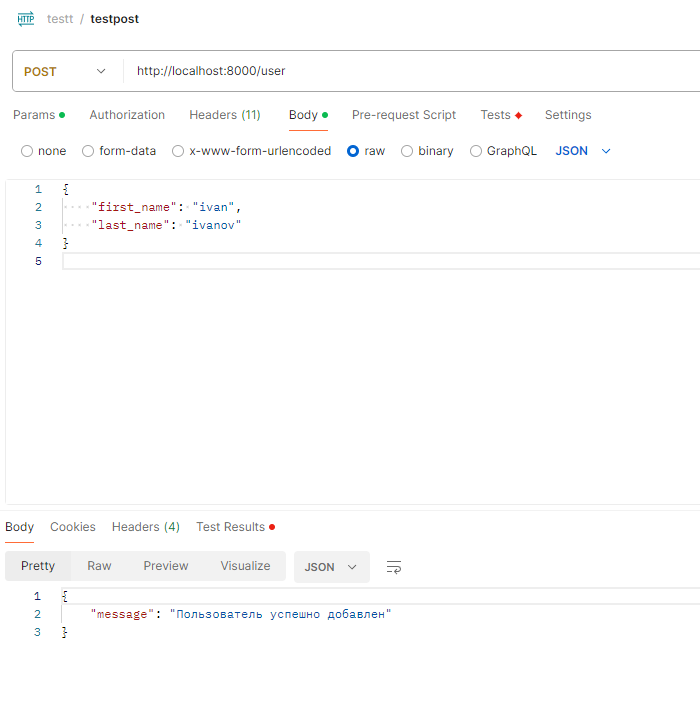


Рисунок 7 – Тестовые данные для Post

Рисунок 8 – Успешное выполнение

API для тестирования методом Get было взято с сайта reqres.

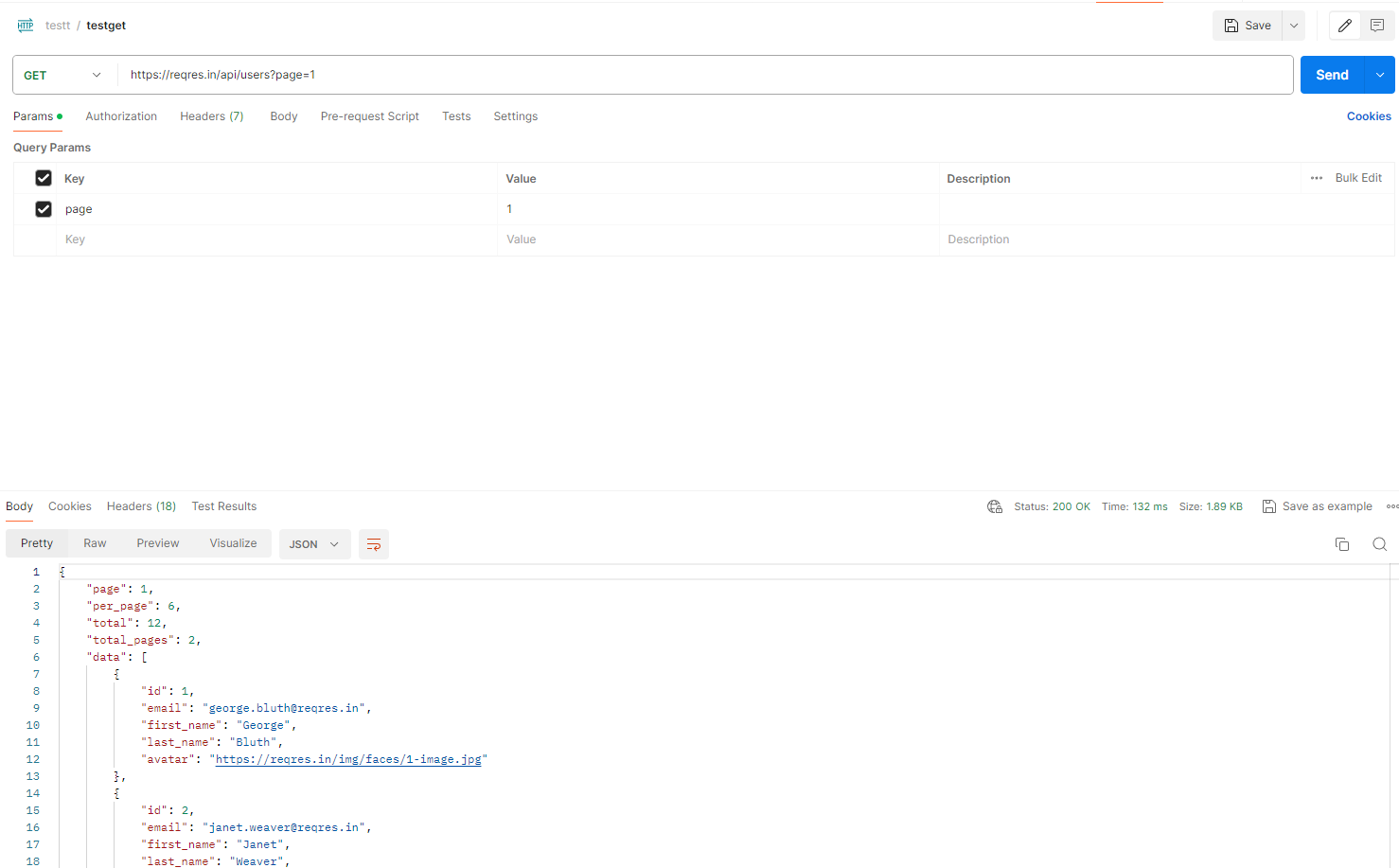


Рисунок 8 – Тестовые данных для Get

## Настройка Linter

Настройка Linter в Python может быть осуществлена с использованием инструментов, таких как Flake8, Pylint или Black.

Flake8 – инструмент статического анализа кода для языков программирования Python, который помогает разработчикам выявлять и исправлять ошибки, следовать стилю кодирования и соблюдать принятые стандарты написания кода. Flake8 проверяет код на соответствие PEP 8 – официальному стандарту написания кода на Python, а также на другие типичные ошибки и неоптимальные ситуации.

Pylint – это еще один инструмент статического анализа кода для языка программирования Python. В отличие от Flake8, который фокусируется преимущественно на стиле кода и соответствии стандартам, Pylint обнаруживает более широкий спектр потенциальных проблем в коде, таких как ошибки синтаксиса, неиспользуемые переменные, несоответствие PEP 8, некорректное использование функций и многое другое.

Pylint также предоставляет детальные отчеты о найденных проблемах и предлагает рекомендации по их устранению.

Black – это инструмент автоматического форматирования кода для языка программирования Python. Этот инструмент задает определенные стандарты форматирования кода, основанные на принципах PEP 8, и автоматически приводит код к соответствию этим стандартам.

Для настройки Linter был выбран pylint. Установить его можно с помощью установщика pip (рисунок 9).

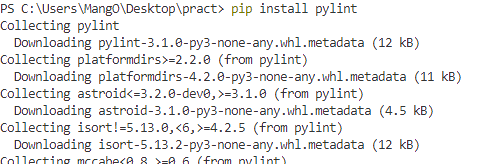


Рисунок 9 – Установка Pylint

Далее создаем файл .pylintrc в корневой директории проекта и добавляем в него конфигурацию.

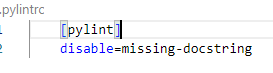


Рисунок 10 – Добавление конфигурации

Отключение предупреждения "missing-docstring" в pylint используется для игнорирования предупреждений о том, что отсутствует строка документации (docstring) для функций, классов и модулей в вашем коде.

Запускаем Pylint (рисунок 11).

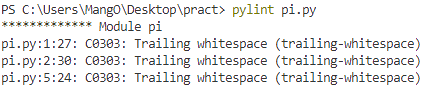


Рисунок 11 – Запуск pylint

ЗАКЛЮЧЕНИЕ

В рамках производственной практики были изучены требования к написанию технического задания для базы данных согласно стандарту ГОСТ. С использованием FastApi был разработан модуль для приема данных извне и их последующей записи в базу данных PostgreSQL.

Также были созданы тестовые данные для проверки функционала методов Post и Get. Прохождение практики позволило углубить теоретические знания, полученные в учебном заведении, и приобрести практический опыт работы с программным обеспечением.

Список используемых источников

1. studbooks.net – Техническое задание на разработку базы данных  URL:  https://studbooks.net/2176887/informatika/tehnicheskoe\_zadanie\_razrabotku\_bazy\_dannyh\_sostavlennoe\_sootvetstvii\_gost34602

2. reqres.in – API для Postman URL: https://reqres.in/