**Интерактивный дашборд с использованием асинхронного программирования в Python**

Введение

Данный проект представляет собой интерактивный дашборд, разработанный с использованием различных подходов к асинхронному программированию в Python: asyncio, threading и multiprocessing. Основная цель проекта — продемонстрировать возможности каждой из этих библиотек для создания динамических веб-приложений с визуализацией данных.

В проекте реализованы три подхода к обработке данных, каждый из которых имеет свои особенности и применения. Дашборд отображает графики, основанные на случайно сгенерированных данных о доходах, расходах и покупках в различных категориях.

# Описание проекта

Цели проекта:

- Обеспечить интерактивность и наглядность данных.

- Демонстрировать использование различных подходов к асинхронному программированию.

- Показать преимущества и недостатки каждого из подходов.

Задачи проекта:

- Генерация случайных данных для визуализации.

- Обновление графиков в реальном времени.

- Сравнение производительности различных подходов.

# Структура проекта

Проект включает в себя следующие файлы:

- main.py - Главный файл запуска, который объединяет все подходы.

- asyncio\_module.py - Модуль для работы с подходом asyncio.

- threading\_module.py - Модуль для работы с подходом threading.

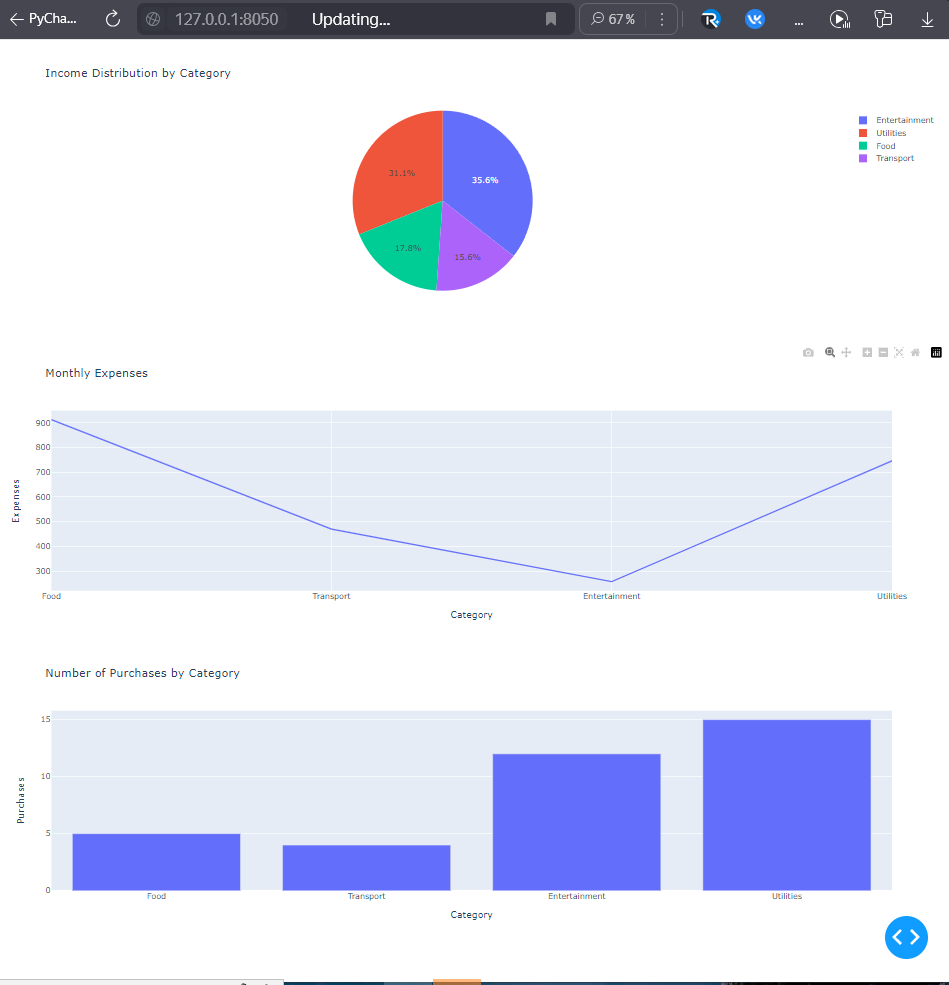
- multiprocessing\_module.py - Модуль для работы с подходом multiprocessing.

- README.md - Документация проекта.

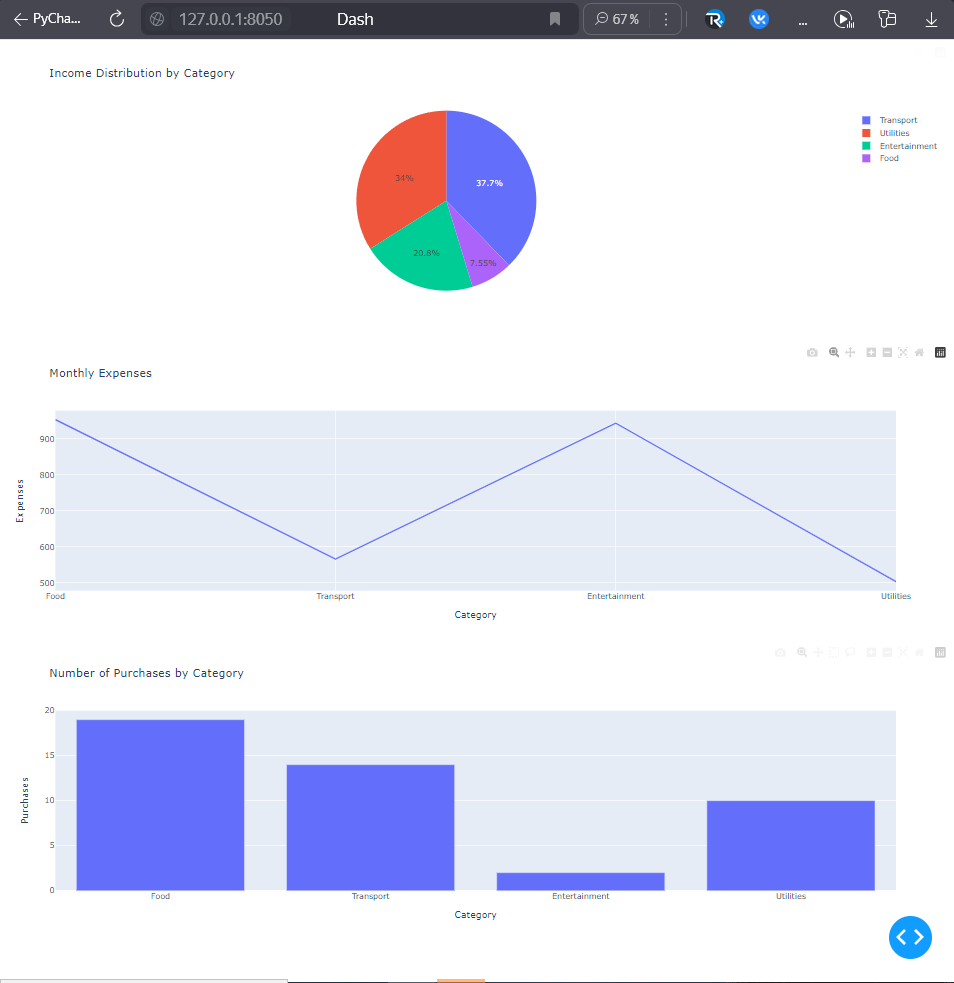
- requirements.txt - Список зависимостей.

Проект включает в себя следующие компоненты:

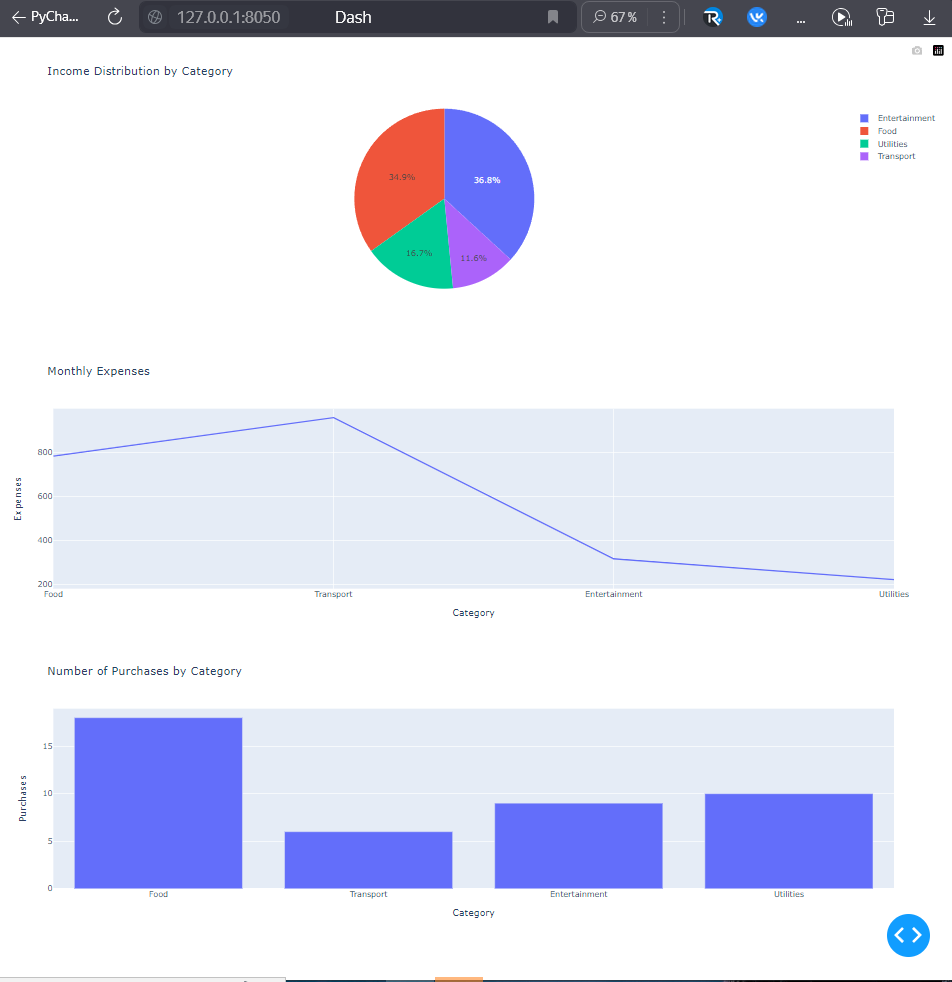
Дашборд на библиотеке asyncio:



Дашборд на библиотеке threading:



Дашборд на библиотеке multiprocessing:



**Основной функционал**

Основной функционал

1.Асинхронный подход (asyncio)

- Генерация данных с использованием asyncio.

- Обновление графиков с помощью асинхронных вызовов.

2. Многопоточность (threading)

- Генерация данных в отдельном потоке.

- Обновление графиков с использованием глобальных переменных.

3. Мультипроцессинг (multiprocessing)

- Генерация данных в отдельном процессе.

- Использование очереди для передачи данных между процессами.

**Сравнение библиотек**

1. asyncio

Преимущества:

- Высокая производительность для асинхронных операций.

- Читаемость кода с использованием async и await.

Недостатки:

- Сложность реализации для новичков.

- Неэффективен для CPU-bound задач.

2. threading

Преимущества:

- Легкость в использовании и быстрота развертывания.

Недостатки:

- Ограничения GIL (Global Interpreter Lock).

- Возможные проблемы с синхронизацией.

3. multiprocessing

Преимущества:

- Масштабируемость с использованием нескольких ядер процессора.

Недостатки:

- Сложность в реализации и управление межпроцессным взаимодействием.

**Заключение**

Проект демонстрирует использование различных подходов к асинхронному программированию в Python для создания интерактивного дашборда. Каждый подход имеет свои преимущества и недостатки.